Answering Questions about Archived, Annotated Meetings

THÈSE N^{O} 4512 (2009)

PRÉSENTÉE LE 30 OCTOBRE 2009

À LA FACULTÉ INFORMATIQUE ET COMMUNICATIONS
LABORATOIRE D'INTELLIGENCE ARTIFICIELLE
PROGRAMME DOCTORAL EN INFORMATIQUE, COMMUNICATIONS ET INFORMATION

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Marita AILOMAA

acceptée sur proposition du jury:

Prof. M. Hasler, président du jury
Dr M. Rajman, directeur de thèse
Prof. N. Castell, rapporteur
Prof. P. Dillenbourg, rapporteur
Prof. J. Moeschler, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE
Suisse
2009

Abstract

Retrieving information from archived meetings is a new domain of information retrieval that has received increasing attention in the past few years. Search in spontaneous spoken conversations has been recognized as more difficult than text-based document retrieval because meeting discussions contain two levels of information: the content itself, i.e. what topics are discussed, but also the argumentation process, i.e. what conflicts are resolved and what decisions are made. To capture the richness of information in meetings, current research focuses on recording meetings in Smart-Rooms, transcribing meeting discussion into text and annotating discussion with semantic higher-level structures to allow for efficient access to the data. However, it is not yet clear what type of user interface is best suited for searching and browsing such archived, annotated meetings. Content-based retrieval with keyword search is too naive and does not take into account the semantic annotations on the data. The objective of this thesis is to assess the feasibility and usefulness of a natural language interface to meeting archives that allows users to ask complex questions about meetings and retrieve episodes of meeting discussions based on semantic annotations. The particular issues that we address are: the need of argumentative annotation to answer questions about meetings; the linguistic and domain-specific natural language understanding techniques required to interpret such questions; and the use of visual overviews of meeting annotations to guide users in formulating questions.

To meet the outlined objectives, we have annotated meetings with argumentative structure and built a prototype of a natural language understanding engine that interprets questions based on those annotations. Further, we have performed two sets of user experiments to study what questions users ask when faced with a natural language interface to annotated meeting archives. For this, we used a simulation method called Wizard of Oz, to enable users to express questions in their own terms without being influenced by limitations in speech recognition technology.

Our experimental results show that technically it is feasible to annotate meetings and implement a deep-linguistic NLU engine for questions about meetings, but in practice users do not consistently take advantage of these features. Instead they often search for keywords in meetings. When visual overviews of the available annotations are provided, users refer to those annotations in their questions, but the complexity of questions remains simple. Users search with a breadth-first approach, asking questions in sequence instead of a single complex question.

We conclude that natural language interfaces to meeting archives are useful, but that more experimental work is needed to find ways to incent users to take advantage of the expressive power of natural language when asking questions about meetings.

Keywords: Natural language interfaces, information retrieval, argumentative annotation, natural language understanding (NLU), Wizard of Oz evaluation, data visualization

Résumé

La recherche d'information dans des archives de meetings est un nouveau champ de recherche dans le domaine de la recherche documentaire qui, depuis plusieurs années, attire une attention croissante. En particulier, la recherche d'information dans des transcriptions de conversations spontanées a été identifié comme une problématique plus complexe que la recherche au sein de base de documents du fait que les discussions tenues au cours de réunions contiennent deux types distincts d'information : le contenu des discussions proprement dit, mais aussi la structure argumentative de ces discussions, comme par exemple les conflits qui ont été résolus ou les décisions qui ont été prises. Pour capturer ce type d'information, les efforts de recherche actuels se concentrent essentiellement sur l'enregistrement de réunions au sein de "Smart rooms", la transcription textuelle des discussions tenues et l'annotation de ces discussions à l'aide de structures sémantiques de plus haut niveau permettant un accès plus efficace à l'information recherchée. Toutefois, il apparaît également que les interfaces standard réalisant une recherche de contenu à l'aide de mots-clés apparaissent comme trop naïves pour une tâche de recherche ou une navigation au sein d'une archive de réunions annotées, le choix de l'interface la plus adéquate pour effectuer une telle recherche est également une question ouverte. Dans cette perspective, l'objectif de ce travail de thèse est d'évaluer la faisabilité et l'utilité d'interfaces de recherche d'information au sein d'archives de réunions permettant aux utilisateurs de poser des questions complexes à propos des réunions archivées et d'identifier, sur la base des annotations disponibles, des épisodes pertinents au sein de ces réunions. En particulier, nous nous intéressons à l'évaluation de l'utilité d'une annotation argumentative pour répondre à des questions à propos de réunions, aux techniques de compréhension du langage naturel, tant linguistiques que sémantiques, nécessaires pour interpréter de telles questions et à l'utilisation de représentations graphiques synthétisant les annotations des réunions pour aider les utilisateurs à formuler les questions adéquates.

Pour atteindre ces objectifs, nous avons tout d'abord produit une archive de réunions annotées à l'aide d'une structure argumentative, puis construit un prototype de module de compréhension du langage naturel capable d'interpréter, à l'aide des annotations disponibles, des questions à propos des réunions. Ensuite, nous avons réalisé deux séries d'expériences pour étudier les questions effectivement posées par les utilisateurs confrontés à une interface permettant d'effectuer une recherche en langage naturel au sein d'une archive annotée de réunions. Pour ces expériences, nous avons mis en œuvre une méthodologie de type "Wizard of Oz" permettant aux utilisateurs d'exprimer des questions de façon naturelle, sans être confrontés aux limitations inhérentes à une technologie reposant sur une approche à base de reconnaissance automatique de la parole.

Nos résultats expérimentaux montrent qu'il est effectivement possible de produire des annotations argumentatives de réunions et de construire un module de compréhension du langage naturel utilisant une analyse linguistique relativement sophistiquée pour répondre à des questions à propos des réunions, mais qu'en pratique, les utilisateurs ne tirent pas systématiquement avantage de ces fonctionalités et se limitent souvent à des recherches sous la forme de simples mots-clés. Nous montrons également que, lorsque des représentations graphiques synthétisant les annotations des réunions sont disponibles, ces représentation sont effectivement utilisées, mais que cette utilisation n'augmente pas de façon sensible la complexité linguistique des questions posées. En effet, dans la plupart des cas, les utilisateurs mettent en œuvre une stratégie consistant à poser une série de questions simples, plutôt qu'un nombre plus réduit de questions plus complexes.

En conclusion, nous mettons en évidence que la disponibilité de techniques à base de traitement automatique du langage naturel dans des interfaces de recherche au sein d'archives de réunions est utile, mais qu'il demeure nécessaire de trouver des moyens permettant d'inciter les utilisateurs à tirer profit de telles fonctionalités lorsqu'ils posent des questions dans le cadre d'une recherche d'information au sein d'archives de réunions.

Mots-clés: Interfaces de langages naturelles, recherche d'information, annotation argumentative, compréhension du langage naturel, méthodologie Wizard of Oz, visualisation de données

Acknowledgments

Writing a PhD thesis is an exciting journey but long and filled with obstacles and unexpected turns. I would not have been able to succeed with this endeavour had it not been for the help and support from a number of different persons. First of all, I would like to thank my thesis director Dr. Martin Rajman for taking me on board and showing such faith and enthusiasm in my work. His guidance has been invaluable throughout all the phases of my research.

An important part of my work relies on conceptualization, implementation and experimentation done in collaboration with other researchers. I would like to give a special acknowledgment to Miroslav Melichar who prototyped the multimodal meeting data retrieval system Archivus and helped me to make necessary modifications in order to perform specialized user studies. He also played a central role in setting up the technical experiment environment and in finding efficient ways to post-process and analyze the recorded experiment data. Without this expert knowledge available, I would still be only halfway through my thesis now. I am also deeply greatful to Agnes Lisowska, who enlightened me on all issues related to user experimentation, such as the design of an appropriate task, how to write tutorials and how to formulate questionnaires. Moreover, Miroslav and Agnes were excellent company during the hours and hours of experimentation that we did together and they truly made the whole experience enjoyable.

Vincenzo Pallotta was of particular help to me in the conceptual part of the thesis. He uncovered the potential of argumentative structuring of meeting data and helped me to apply his theoretical annotation schema to my experimental work. I am also very grateful to him for the useful pointers to relevant literature and different angles on the problems I was addressing, which lead to new insights and progress. I also want to thank Hatem Ghorbel and Violeta Seretan for taking part in the time-consuming work of annotating and categorizing meeting data and analyzing the outcomes.

I am very thankful to the members of my thesis committee: Prof. Pierre Dillenbourg, Prof. Jacques Moeschler and Prof. Nuria Castell, for their positive feedback and helpful comments about final improvements of my thesis.

Finally, I want to thank my husband Jens Ingensand who supported me in every conceivable way, including professionally, domestically and morally, to ensure that I completed my thesis in time. I feel very lucky to have him at my side, and I hope I can find a way to return the favour one day. Thank you!

This work was funded by the Swiss Interactive Multimodal Information Management (IM2) project and the Swiss National Science Foundation grant number 200021-116235.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Retrieving information from meetings

In our daily lives we are surrounded by information. It is spread through newspapers, television, books, and in recent years also widely through the internet. The advances in information technology has made it possible to access information from anywhere at anytime. The technologies in turn have created new needs for storing, organizing and making data available for search. But aside from all this recorded, written, or otherwise physical information, we are also dependent on human-human communication to exchange information. Through dialogue we transfer knowledge that can be necessary for solving problems and completing tasks. In many situations the expert knowledge that is exchanged orally can play a more important role in achiving goals than any written documents. For example, in 1969 when NASA made the first moon-landing, the preparations for that mission were enormous, and innumerable meetings were held to discuss issues related to the execution of the mission. Today, NASA is planning a similar voyage to Mars, but the project has been slowed down by an unexpected problem: much of the expert knowledge needed to enable this venture has been lost, either because the information that was communicated during the meetings in 1969 were never captured in writing, or because the notes from those meetings have disappeared. It is now challenging work to reconstruct that knowledge.

In general, the information that gets exchanged through discussions between people can be important for the advancement of projects, assignment of responsibilities and decisions on future actions. From this perspective, it is surprising that such valuable information often exists only in those persons' memories who participated in the discussion, and is not otherwise traceable. The exception is if a discussion was held formally in a meeting. Then someone may have written down the minutes. Or if two persons communicated through computer technology, e.g. live-chat software, then the conversation may have been registered in a chat history. And if a discussion was held publicly, it may have been recorded by media and made available online. But in cases where conversations have not been captured or documented in any way, one has to rely on the persons who were present in the discussion, to reconstruct what was said. Human memory is unreliable and the information that was communicated in conversations often gets partially lost or modified as time passes. Due to this deficiency, and the fact that computer and information technologies develop rapidly, a growing interest has been directed towards recording and storing meeting discussions to make them available for future retrieval, either for the persons who participated in the discussion; or when it is relevant, for external persons who want to know what was discussed.

The most straight-forward way of extracting information from recorded conversations is to browse them from the beginning to the end. A more efficient approach, however, is to directly access the parts where important things were said. But how to characterize 'important' parts of a conversation is not evident. The main reason is that the task of retrieving information from recorded conversations is largely unfamiliar to average computer users. Currently, when a person wants to find out what was said in a discussion that they did not attend they ask another person who was present. The type of questions that they ask can be very intuitive and simple for a human to answer, for example "What decisions were made?", but very complex if the question is asked to a computer. The complexity arises from the fact that questions about discussions can refer to many dimensions of the discussion, not only the content itself, i.e. what precise words that were said. Other dimensions are the argumentative dimension, for example if there were conflicts between the participants and how they were resolved; or the activity dimension, for example if a person was making a presentation or if someone was drawing on the whiteboard.

The most standard approach to information retrieval in any domain currently is content-based search, i.e. matching keywords in the query with terms in the searched documents. It has become standard because it is technically the easiest to implement, and users have adapted to the technological limitations by learning to choose keywords that are most likely to deliver the documents that they want. Searching conversations with keywords, however, is problematic. Recorded conversations do not have the same properties as written documents. Except for the different information dimensions that were mentioned above, speech is spontaneous and unstructured; there are many disfluencies, interrupted sentences, and occurrences of overlapping speech. Sentences often do not contain the explicit terms that reflect the topic that is being discussed, because participants assume certain implicit, shared knowledge when speaking. Minimizing redundancy is a linguistic phenomenon that humans apply naturally when speaking. Moreover, conversations are multimodal. Gestures, gaze and tone of voice can communicate moods, attitudes, irony and jokes, which in turn add new meaning to what was said. In brief, the information in conversations is represented by much more than the words themselves and retrieving the important parts from conversations with content-based search alone is not the solution.

1.2 Open research issues

The challenge of developing information retrieval systems for conversational data has been acknowledged by the emergence of large-size research projects (e.g. IM2[1] , AMI[2] , ICSI Meeting Project[3]), and workshops (e.g. SSCS[4], MLMI[5]) focusing on this specific problem. There are at least three broad research issues being addressed.

The most fundamental research issue is the transcription of multi-party discussions. On one hand, a method for human expert transcribers needs to be designed, that takes into account all the conversational phenomena in human communication, such as overlapping speech, laughing, mumbling, hesitating, etc. The question here is how much of what happens in discussions, including multimodal messages, needs to be transcribed to preserve as much as possible of the original information in conversations. The speech group at ICSI, for example, addresses the issue of rich transcription of natural and impromptu meetings (Janin *et al.*, 2004). On the other hand, in order for conversation retrieval to be realistic on large scale, transcriptions have to me made automatically. Automatic speech recognition (ASR) is a well-known and extremely difficult problem that has challenged engineers for years. Within the scope of meeting transcription, research focuses on particular issues related to the transcription of spontaneous spoken conversation as opposed to well-pronounced clean sentences. Research activities in this area take place for example at the research institute IDIAP[6] in the framework of the Swiss IM2 project. The annual MLMI workshop is one of the forums where progress in this work is reported.

At a higher level of data processing, an open research question is how to enhance transcriptions of conversations with structure and annotations to enable more efficient search of this data. On one hand, it remains unclear which annotations are the most useful for retrieving important parts of conversations. Some types of annotations that have been proposed are topic segmentation of discussions, and labelling of the segments with representative keywords or concepts e.g. (Galley and Mckeown, 2003); dialogue act annotation of utterances e.g. (Shriberg *et al.*, 2004); and argumentative segmentation and categorization of the discussion (Pallotta *et al.*, 2004; Verbree, 2006). However, few evaluations have been made about how real users would want to exploit such annotations when searching in conversations. On the other hand, just like with transcription, in order for higher-level discourse annotations to be realistic in large scale conversation retrieval systems, they need to be made automatically. For many of the proposed annotation schemas, automatic annotation algorithms do not yet exist.

At the interface level, there are many possibilities for how to access conversational data. In particular, if discussions take place in formal meetings, there are often documents involved, either distributed among the participants or projected on a wall. Important parts of the discussions may refer to such documents, and it may be relevant to retrieve not only the conversation itself but also these multimedia documents that are being referred to. The issue of

[1] http://www.im2.ch/

[2] http://www.amiproject.org/

[3] http://www.icsi.berkeley.edu/Speech/mr/

[4] http://hmi.ewi.utwente.nl/sscs

[5] http://www.mlmi.info/

[6] http://www.idiap.ch/scientific-research/projects/audio-processing

interface design is addressed by the IM2 and AMI projects by developing and testing various prototypes of multimedia meeting data retrieval and browsing systems, commonly known as 'meeting browsers' (Bouamrane and Luz, 2007). Here the main focus is on developing optimal search and browsing techniques for particular types of media: the meeting transcription, original recording, referenced documents or meta-information, e.g. where and when a discussion took place.

The research on conversation data retrieval is driven by the potential benefits that it can provide for businesses and institutions where meetings are a central part of their activities. The vision is that meeting data retrieval systems will contribute to more efficient team work, more productive meetings and higher quality in project outcomes. When sufficient progress has been made on the above open research issues, conversation retrieval may become a part of professional working routines.

1.3 Contributions of this thesis

In this thesis, the main issue being addressed is the interface between the user and the meeting database. The goal is to reduce the gap between natural human-human approaches to answering questions on conversations, and the current human-computer approaches to information retrieval. The question we want to answer is: what happens when natural language is introduced as a search method in the interface? Is it more efficient, satisfying, or natural for search than plain keywords? To answer this question, we propose to build and evaluate a system that distances from content-based search technologies and uses natural language as its main search modality. Concretely, the work that we have done is:

- Annotate conversations with higher-level structures to enable users to ask questions about conversation on a semantically higher level, like in the human-human approach. Here we focus on the argumentative dimension of discussions and assess the difficulty of annotating meeting discussions with argumentative structure. Moreover, we perform a set of user experiments to find out what annotation users exploit in real life when searching in meetings. We contribute to the open research issue on the usefulness of higher-level annotations by analyzing to what extent users go beyond content-based search when argumentative annotations are available.

- Develop natural language understanding (NLU) techniques for interpreting questions on structurally annotated conversations, so that complex questions asked by users can be answered by the system. We use existing NLP techniques to obtain a linguistic analysis of the question, and domain-specific semantic interpretation techniques that exploit the linguistic analysis. To our knowledge it is the first deep-linguistic natural language understanding engine for questions about meeting discussions. We contribute to the issue of accessing meeting data by providing the NLU technology needed for developing natural language interfaces to meeting archives.

- Design a controlled laboratory-experiment to be able to make user evaluations with a natural language interface to meeting archives. We have chosen a simulation method known as the Wizard of Oz method, a standard method for evaluating telephony-applications. Our contribution is the extension of this method to language-enabled multimodal graphical user interfaces.

- Introduce visual overviews of meeting discussions, as aid for searching and browsing structurally annotated meetings, and perform user experiments to evaluate the value of these overviews for enhancing the overall task. Our first contribution is the design of a new type of conversation graph that visualizes three dimensions of a discussion: what topic was discussed, who made argumentative contributions, and what type of contributions those were. Our second contribution relates to the issue of appropriate interface design fr accessing meeting data by comparing how topic overviews and conversation graphs influence querying and browsing.

The core of this work is to show that a natural language query engine to archived meetings is technically feasible, useful and appreciated by users. In terms of feasibility, we want to show that the argumentative annotation of conversations allows for answering relatively complex questions about conversations, and that the natural language understanding of such questions can be highly reliable, if questions are interpreted with a combination of linguistic and domain-specific techniques. In terms of usefulness, the goal is to show that natural language provides more efficient search and browsing capabilities than menu-based graphical user interfaces, and that users exploit meeting annotations to a higher extent when interacting in natural language, in particular when the task is enhanced with a visual overview of the available annotations. Finally, in terms of appreciation, our goal is to show that users prefer natural language-enabled graphical user interfaces over standard graphical user interfaces for retrieving answers to questions from archived meetings.

1.4 Structure of the thesis

The work in this thesis is structured into five chapters.

Chapter 2 outlines the state of the art in related research fields. First, we describe work on natural language and multimodal interfaces, applications in which natural language is used for searching for information, and various techniques for natural language understanding of questions. Then we give an overview of experimental evaluation of natural language interfaces, including the standard evaluation methods and the different aspects of natural language interaction that can be evaluated with these methods.

Chapter 3 describes the work on the technological or implementation aspect of meeting data retrieval systems. First, we give an overview of the type of questions that are typical about archived meetings, and discuss the system design implications for answering those questions. Then we propose a system-architecture for a natural language query engine to meeting archives, focusing on the requirements for the database and natural language understanding modules. We present the results of a small-scale argumentative annotation study, assessing the difficulty of annotating conversations with argumentative structure. We also provide a detailed description of the linguistic and domain-specific techniques used for interpreting questions about discussions, and the evaluation of the final natural language understanding module that implements those techniques.

Chapter 4 addresses the usability aspect of natural language-based interfaces to meeting archives. We describe the multimodal user interface, Archivus, which we used as a case study for user evaluations. Then we describe the experimental evaluation framework, and how the Wizard of Oz method was extended to multimodal interfaces. We provide the concrete objectives of the user study as a set of research questions and hypotheses about what types of annotations users exploit when they search, how complex their questions are linguistically, and how willing they are to ask questions in natural language when they have the option to search with menus. We then provide the experimental results for these questions and hypotheses, as well as conclusions.

Chapter 5 addresses the task of answering questions on annotated meetings. Here we describe the design of a new type of visual meeting overview, conversation graphs, that we propose as visual aid for querying and browsing archived meetings. In this chapter we specify a set of research questions and hypotheses about how conversation graphs can influence the task, e.g. by incenting users to exploit the argumentative annotation, ask more complex questions, and browse search results more systematically. Experimental results on these questions, and conclusions, are provided.

Chapter 6 draws general conclusions about the significance of this work to real-world contexts, and how our findings contribute to the progress in developing future meeting data retrieval systems.

2
State of the art

The work in this thesis draws from the broad research field of natural language interfaces to computer systems, and applies findings in the field to the application of natural language interfaces to archived, annotated meetings. In 2.1 we first present different ideas about why and how natural language should be used when interacting with computers. We then continue in 2.2 with an overview of applications where natural language is used for information search. In 2.3 we review current natural language understanding techniques for computing the meaning of natural language questions. Finally, in 2.4 we provide a description of methods for evaluating the performance and usability of natural language interfaces.

2.1 Natural language interfaces: what are they good for?

2.1.1 Limitations of graphical user interfaces

Graphical user interfaces (GUIs) made a breakthrough with the introduction of the PC and are nowadays the most commonly used interfaces in most computer applications. People are so used to interact with keyboard and mouse, that some, e.g. Shneiderman (2000), argue that natural language interfaces (NLIs) can never be as efficient as GUIs, except in limited special-case situations. There are however many arguments for why natural language interfaces, especially speech interfaces, can become important alternatives to GUIs in the future. Some arguments are:

- Interaction devices are progressively becoming smaller and making efficient GUIs less feasible. Speech interfaces can be used with any size of devices and can be scaled to any number of concepts and operations (Rosenfeld *et al.*, 2001; Leidner, 2005; Katz *et al.*, 2007).

- Some tasks require using ones hands and/or eyes. In such contexts, GUIs cannot be manipulated directly, but humans can easily interact with speech while performing other tasks (Bernsen and Dybkjaer, 1999; Cox *et al.*, 2008).

- A new GUI requires some amount of training from the user to learn its functionalities. Language-based interaction can be a valuable additional modality as it adds considerable flexibility (Walker *et al.*, 1998; Olsen, 1999).

- GUIs do not let users communicate in ways that they naturally do with other human beings and therefore reduce their possibilities to rely on intuitions derived from human-to-human interaction (Sidner, 1997; Edlund *et al.*, 2008).

As pointed out by Shneiderman and Plaisant (2009), there is a fundamental difference in the nature of GUIs and NLIs that makes each of them appropriate for different types of tasks. GUIs are suitable for displaying limited amounts of rich information in a structured way. Metaphorical objects such as buttons, menus and scrollbars allow users to learn quickly, because once they get familiar with the basic alphabet of interactive behaviors they know how to interact with a wide variety of applications. However, the interaction with a GUI is limited to the objects visible on the screen. If the task is complex, the designer must either keep the interface simple by providing basic operations but then force the user to go through long sequences of commands to perform the task, or the interface can provide higher-level commands that perform the task, but these commands then typically need to be clustered into task classes, requiring significant training from the user. Natural language, on the other hand, is a free form of interaction. Language enables users to state what they want in their own terms, leaving the complexities of the task to the system. For instance, language allows users to select items by directly referring to them, which is more efficient than going through a list of options. Similarly, language enables users at any given time to talk about entities not visible on the screen. The integration of natural language with graphical interfaces is a research topic that has long been identified as important but has not yet yielded a good understanding of how this integration should be done.

2.1.2 Limitations of natural language interfaces

Even before computers came into existence, there was a dream of creating machines that could understand and speak natural language. Books and films from the 60s envisioned robots that would work side by side with humans, and later Hollywood productions such as *Artificial Intelligence AI* and *I robot* went further and addressed the emotional relationship between machines and humans. The general interest in human-like machines is further emphasized by initiatives to set up competitions where awards are offered to those who manage to come closest to satisfying the Turing test, i.e. to create computer programs that are so human-like that they in fact become undistinguishable from a human. One such competition is the Loebner Prize[1], which has been hosted since 1991. However, research on natural language interfaces has made it evident that designing general purpose natural language interacting computers is too ambitious. Computers cannot be made to understand natural language in a broad sense. Language is subtle: meanings are context-dependent; the grammar is full of exceptions, and human-human communication is coloured by emotion and implicit messages. Instead, research on natural language interfaces has

[1] http://www.loebner.net/Prizef/loebner-prize.html

moved towards creating specialized computer programs for limited applications. Some examples are dialogue systems for time table consultation (Aust *et al.*, 1995), translation systems for medical doctors and patients who do not speak the same language (Rayner *et al.*, 2008), or voice-control of gadgets in the car (Sporka and Slavik, 2008). To fuel research and development of such interfaces, tools and technologies are made available to facilitate the implementation. Bird *et al.* (2008) provide an open-source NLP toolkit that contains code supporting dozens of NLP tasks. Cimiano *et al.* (2007) propose tools for porting existing natural language interfaces to new domains without requiring any computational linguistics expertise. Frost (2006) provides a survey of programming languages that are most suited for effortless development of natural language interfaces.

One of the drivers behind the development of NLIs has been to replace GUIs in situations where language interaction is more practical or efficient. However, in recent years it has become increasingly hard to identify such situations. GUIs have improved enormously since research on human-computer interaction (HCI) showed that successful interface design starts from a user-centered perspective with analysis of users needs, rather than a system-centered perspective with focus on finding solutions to complex computational problems. Todays GUIs are able to display information in compact, intelligent ways, and provide easy access to objects through direct manipulation. When observing the long-term development, GUIs have often won over natural language interfaces, even when the natural language interface originally yielded positive feedback (e.g. Shneiderman 1980; Dekleva 1994). Menus are perceived as easier to use than natural language interfaces because they show what the choices are (Hasan and Ahmed, 2007; Shneiderman and Plaisant, 2009). Nevertheless, there are situations in which menu-selection can be tedious, or inherently difficult to use, as in the case of interfaces to relational databases (Jagadish *et al.*, 2007). In such situations, natural language interfaces are foreseen to become useful complements to GUIs.

An important debate within the NLI community is whether the most natural and desirable way for humans to interact with computers is with a free form of conversation or with a restricted, predictable style of interaction. Past and current research on conversational dialogue with computers (Seneff, 1992; Allen *et al.*, 2001b; Traum *et al.*, 2005; Skantze, 2005; Edlund *et al.*, 2008) relies on the assumption that users may feel more comfortable with an interface that possesses some of the characteristics of a human agent, and that studying human-human dialogue can provide valuable insights for the design of conversational interfaces (Bernsen *et al.*, 1996; Zue and Glass, 2000; Andre, 2003). Considerable work has been done in the field of linguistics to achieve models of human-human interaction, the most fundamental being the speech act theory (Austin, 1962; Searle, 1969) and the dialogue act theory (Bunt, 1981), but they are very difficult to apply to robust computational mechanisms for dialogue processing, and few systems operate on the speech act level (Allen *et al.*, 2001a). It has also been observed that human-human dialogues often change direction in a way that does not contribute directly to goal-directed problem solving, and that the lack of a precise model in human-human communication makes it counter-productive to use it as a basis for the design of goal-oriented conversational systems (Thomson and Wisowaty, 1999). Instead, new models that better conform to computational implementation and that are more representative of problem-solving dialogue, such as the Issue under negotiation model (Larsson, 2002) have been proposed to achieve more practical human-like dialogue systems.

The other view in interface design research is that the intrinsic difficulty of the speech recognition and NLP tasks make it unlikely that free conversational interaction will ever be as efficient as interaction with a controlled language, because it requires large grammars and lexica, and efficient clarification, confirmation and error-correction mechanisms (Rosenfeld *et al.*, 2001; Shneiderman and Plaisant, 2009). Furthermore, “natural” interaction does not have to imply “human-like” interaction. Naturalness is mainly about how to build the right interface. The key requirement is that it should be easy for users to determine what objects and actions are appropriate. An approach proposed by Tomko *et al.* (2005) is to mimic the principle of GUIs, i.e. to identify a small set of universal interaction “primitives” (system prompts and response types) that recur in many applications and, in fact, constitute the great majority of turns in these applications. Once users have learnt how to use one application, they know how to use others.

Little empirical study has been done to support either view on how natural language interfaces should be designed, but some results show that users are generally more comfortable with a less flexible and system-driven interaction even if expressing a request in a direct way is more efficient in terms of number of steps to solve the problem. Users prefer systems with predictable behavior (Walker *et al.*, 1998). However, other studies show that users who are used to tool-like interfaces with little or no natural language capabilities still prefer a human-like interaction once they have experienced it (Chai *et al.*, 2001; Qvarfordt *et al.*, 2003; Edlund *et al.*, 2008). In short, there is uncertainty, both among researchers and developers, about how natural language interfaces should work in order to make the interaction efficient and agreeable to users. There is a general agreement that more well-grounded experimental evidence is required to shed light on this issue.

2.1.3 Multimodal interfaces

Due to the different strengths and weaknesses of direct manipulation and natural language, it has been acknowledged that these two modalities could complement each other by being accessible simultaneously in a multimodal graphical user interface (Grasso *et al.*, 1998; Bernsen, 2001; Andre, 2003). There are at least two definitions of “multimodal interaction” in this context. The first is that the user can switch between modalities, and choose the preferred modality for a given action, for example to choose a value from a menu with direct manipulation, or to say the value in natural language. The other is that the modalities are used synchronously and that the input from one complements or disambiguates the input from the other, for example when the user says “Put that there” and points at two locations A and B. In an early study by Walker and Whittaker (1989) that focused on the first definition of multimodality, a set of natural language functionalities were identified as being useful complements to menus in the task of retrieving information from a database. Some of the functionalities were: sorting data with self-chosen criteria (e.g. by month or by corporation), expressing requests with negation ("*All customers except those who placed an order at Megastore*"), and coordination of multiple requests in one query ("*List sales to Megastore AND JumboShop*"). The study showed that although users had been trained to use the available set of functionalities, in practice they only used a subset of them, and the extent to which they used them depended on whether the user was a persistent or naive one. Later studies gave similar results (Cohen, 1992; Sturm *et al.*, 2002). The underlying implication is that building a multimodal system does not necessarily mean that users will take advantage of the unique properties of each modality when interacting with the system. When designing multimodal interfaces, care has to be taken to how modalities should be integrated to achieve user acceptance (Bernsen, 2001; Oviatt *et al.*, 2004).

Oviatt (1999) clarifies the issue of why multimodal interfaces are not by default accepted by users. She lists 10 myths about multimodal interaction that has influenced the development on multimodal interfaces in the past. Two of those myths are that speech is the primary input mode in any multimodal system that includes it, and that enhanced efficiency is the main advantage of multimodal systems. In reality, the usefulness of interacting multimodally depends highly on the task. Accessing multimedia data (Andre, 2003) and manipulating spatial data (Rauschert *et al.*, 2002; Andr *et al.*, 2004) are two such tasks where natural language has been shown to enhance the interaction with a GUI. More recently, user studies have been made to assess the usefulness of multimodal interfaces for the web (Stanciulescu *et al.*, 2005; Neto *et al.*, 2009), and development-toolkits have been created for developing, deploying and evaluating web-accessible multimodal interfaces (Gruenstein *et al.*, 2008). The general research trend is to introduce natural language as a modality in GUI-based information search tasks. In our work, the task at hand is searching in archived, annotated meetings; and one of our goals is to evaluate the appropriateness of a multimodal natural language and direct manipulation interface for this specific task. This work overlaps not only with research on multimodal interfaces, but also with natural language interfaces to information search applications. The next section describes the field in more detail.

2.2 Natural language for information search tasks

2.2.1 Natural language interfaces to databases

A natural language interface to a database (NLIDB) is a system that allows users to access the contents of a database by asking questions in natural language (Androutsopoulos, 1995). The first prototypes appeared already in the late 60s and early 70s (Woods *et al.*, 1972). The reason why natural language was so interesting for this specific application was that non-experts found it extremely difficult to access databases with formal query-languages such as SQL. Even today, the user-friendliness of interfaces to databases remains an issue (Jagadish *et al.*, 2007).

Research on NLIDBs boomed in the 80s. Commercial products emerged and were sold to businesses where they were used on a daily basis. Some of the most successful were INTELLECT, Q&A and English Query. However, when GUIs were invented, research and commercial products focusing on NLIDBs gradually died out. The performance of NLIDBs simply did not keep up. There were several problems: 1) the linguistic coverage of the system was not obvious to users. The system could sometimes perfectly answer one question, but fail to answer an almost identical one. 2) The nature of failures could not be distinguished. If a question provided no answer, the user did not know if the problem was due to the linguistic scope of the system or the conceptual scope of the database. 3) Natural language questions were ambiguous. The answer was often not what the user expected.

For a long time the general opinion was that the problems associated with NLIDBs are too difficult to solve, and that NLIDBs will never be truly practical for accessing databases. However, in recent years, as more and more non-expert users access information with web browsers, Smart phones and other devices; and as database interfaces still remain difficult to use, new efforts are being made to improve the performance of NLIDBs. Popescu *et al.* (2003) propose an implementation that reliably maps natural language questions to SQL for a specific class of questions that they define as “semantically tractable” and which in fact constitute the majority of questions to NLIDBs. Many researchers propose conversational dialogue to disambiguate and correct questions that map to incomplete SQL-queries, e.g. Ioannidis and Viglas (2006) and Boye and Wiren (2008). Keyword-search techniques originally designed for retrieving unstructured data such as textual documents have also been tested for NLIDBs (Agrawal *et al.*, 2002; Bhalotia *et al.*, 2002; Chen *et al.*, 2009). To date there are no definite solutions to NLIDBs and there are many unsolved research problems. On the other hand, there is much research going on to solve problems in the field of question answering (QA) on unstructured data. Collaboration between these two fields may lead to new progress.

2.2.2 Question answering

Question answering (QA) is defined as the task of automatically answering a question posed in natural language (Voorhees, 2001). There are two main directions in QA. The first is open-domain QA where the goal is to answer questions about any topic, using the web as information source, and retrieving sentences from documents that contain the answer (Prager, 2006). Some QA systems found online are START[2], Powerset[3] and Ask[4]. The second direction in QA is restricted-domain QA (Molla and Vicedo, 2007) where special-purpose techniques are used for answering questions in closed domains, such as medicine (Niu and Hirst, 2004), or geography (Ferres and Rodrguez, 2006).

In open-domain question answering, questions are classified according to types, which determine the techniques most appropriate for retrieving the answer.

[2] http://start.csail.mit.edu/

[3] http://www.powerset.com/

[4] http://www.ask.com/

For example, factoid questions ("When did Mozard die?") are answered by matching the lexical and syntactic constituents of the question with sentences in free text that have a similar structure, but which are formulated as statements ("Wolfgang Amadeus Mozart died in 1791"). The same approach is used for yes/no questions ("Did Bell invent the telephone?") and some wh-questions ("Who was the singer in The Ramones?"). More sophisticated techniques are required for questions where the answer is not likely to have any syntactic similarity with the question, e.g. why-questions ("Why did the US attack Iraq?"), questions about properties ("What type of bridge is the Golden Gate Bridge?") or list-questions that can provide hundreds, or even thousands or answers ("Which hotels are there in Florence?"). To rank the performance of state-of-the-art open-domain QA prototypes, the TREC conference hosts a yearly QA track, where each prototype is evaluated against a set of pre-selected questions (Voorhees and Buckland, 2007).

In restricted-domain question answering, techniques vary in terms of how the data is structured or exploited to maximize precision in extracting answers. Question answering on speech transcripts (QAST) (Turmo *et al.*, 2009) deals with issues such as the robustness of QA techniques to automatically recognized speech data with high word error rates (Comas and Turmo, 2009). Agichtein *et al.* (2007) addresses QA on web documents that have implicit structure, i.e. where data is organized in tables or lists but lack information about the schema. The authors show that such structures can be exploited to answer classes of questions that cannot otherwise be answered with current techniques in open-domain QA. Katz *et al.* (2007) propose to improve answer extraction by enhancing unstructured data with natural language annotations, i.e. computer-analyzable collections of natural language sentences and phrases that describe the contents of various information segments. The goal is to bridge the gap between sentence-level text analysis capabilities and the full complexity of unrestricted natural language text. The intuition is that answers cannot always be extracted from a single sentence, but a representative sentence can provide the pointer to a paragraph that provides the complete answer. Instead of the data-driven approaches to restricted-domain QA taken by many, Hallett *et al.* (2007) addresses the problem of formulating appropriate questions to limited-domain knowledge systems. Instead of making users go though training in question composition, the authors propose a conceptual question authoring technique that allows users to make complex questions and successfully retrieve answers to them.

As QA becomes more and more targeted on restricted domains, and as the domain knowledge is often structured in some way, the borders between QA and NLIDBs start becoming fuzzy. In particular, with the gradual development of the Semantic Web, i.e. web content that is annotated using semantic ontologies, natural language querying of such semantically enhanced data is becoming the new research challenge.

2.2.3 Querying semantically annotated data

One of the motivations of enhancing web content - or any textual data - with semantic annotations, is that it allows for retrieving information based on what the document is about, rather than what the document says (Henstock *et al.*, 2001). For example, queries that describe the type of information that the user wants (e.g. “Shops that sell baby clothes”) can be matched to relevant websites based on semantic annotations of such websites as being of type “shop”, selling products of type “clothes” and focusing on customers of type “babies”, instead of retrieving websites that contain the terms “shop” and “baby clothes”. The same reasoning applies to question answering. Segments of documents that represent answers to questions can be extracted based on what they talk about, rather than what words they contain.

The need for natural language interfaces to semantically annotated data arises from the fact that computer-analyzable ontologies that encode this expert-knowledge about the content of documents is difficult to understand for end-users. Natural language interfaces to such ontologies represent the most intuitive way of exploiting semantic annotations (Lopez *et al.*, 2005; Kaufmann *et al.*, 2006; Ramachandran and Krishnamurthi, 2009). Techniques required for interpreting questions in open-domain QA based on semantic web ontologies are examined by Tartir *et al.* (2009).

There are however other contexts than web search in which data can be enhanced with semantic annotations to enable efficient natural language querying of the data. One such context is the access to multimedia data, in particular video recordings. Linckels *et al.* (2007) propose a semantic search interface to recorded university lectures. Students can ask questions in natural language and retrieve few and pertinent learning objects, e.g. short multimedia documents. The pertinence of the learning objects is determined by interpreting questions against a domain ontology describing the lectures. A more open-domain approach to natural language querying of video recordings is to annotate videos based on general ontologies describing objects, the objects spatial properties, and activities that occur in video frames (Erozel *et al.*, 2008). However, the more general the ontology, the less expressive power it provides for answering questions. Semantic annotations are generally considered more meaningful when they encode domain-specific knowledge.

In the domain of multimedia meeting data retrieval, video recordings are of central importance. Meetings are recorded in order to enable efficient search of specific issues that were discussed. Semantic annotations of videos are highly relevant in this case, but the more common approach is to first transcribe the spoken content in the video into text, and then enhance the transcription with semantic annotations. This thesis elaborates on the exploitation of such semantically annotated transcriptions and contributes to the research by assessing the feasibility and usefulness of enhancing meeting discussions with semantic annotations to retrieve answers to questions. The feasibility is determined by whether questions can be understood correctly with respect to the ontology (domain-model) of meeting discussions. The next section reviews linguistic and domain-specific techniques for understanding natural language questions.

2.3 Natural language understanding

One of the important current challenges for the design of natural language interfaces is to find ways for computers to "understand" natural language. Language is a very complex communication protocol among humans, and to develop computational theories that consider all the aspects of language (structure, meaning, intentions, etc.) requires collaboration between different research disciplines such as linguistics, computer science, philosophy and cognitive science.

The term Natural Language Understanding (NLU), when used in the field of computer science, refers to the task of building computational models of human language that will enable effective human-computer communication (Allen, 1995). There are two types of NLU applications: (1) text based applications that involve the processing of written text, such as books, e-mails, reports, and so on, and (2) interactive applications that naturally involve spoken language, but also written language if the interaction happens with a keyboard. This second type of NLU applications are the focus of this thesis.

What it means to say that a computer understands human language depends on the complexity of the specific application. For systems that are designed for simple and straight-forward tasks, the understanding can be considered as naive because the system may only be able to process specific utterances and keywords, for instance a predefined set of commands. Other systems that are designed for more complex tasks such as problem solving, negotiation or question answering require significantly more sophisticated computational models of language understanding. In this section we describe three of the existing approaches to NLU (template matching, knowledge-based understanding with logical forms, and grammatical relations) that we find to be either the most common ones or the most relevant to our present research. We also give examples of their applications to different language understanding systems.

2.3.1 Template matching

Template matching (or concept spotting) is the simplest form of NLU, highly appreciated for its relative ease of development and for not requiring a wide linguistic competence on the side of the developer (Ward, 1989; Wang and Acero, 2005). Typically the task is very specific and relies on special-purpose techniques exploiting the domain structure (Hacioglu and Ward, 2001; Eun *et al.*, 2005). Such techniques, though limited to the specific task, often produce more successful systems than the ones based on general-purpose techniques, where each sentence needs to be completely parsed and interpreted before information can be extracted. The basic idea with such limited domain systems is that you can specify simple patterns that indicate key pieces of information in the domain. This information is then used to fill in templates that represent the task. For instance, in the train schedule domain, the preposition 'from' usually indicates the departure location and can be expressed with a pattern, such as:

from <CITY> → Departure-City: <CITY>

where the left hand side of the rule indicates what is to be spotted in the textual input and the right hand side defines the associated attribute-value pairs to be produced. More complex patterns can be designed to deal with multi-word expressions and phrases, but the input must then be parsed at least to identify occurring noun phrases (Cheadle and Gamback, 2003). Partial parsing techniques can be used if necessary (Kaiser *et al.*, 1999).

Systems that rely on template matching for the understanding of natural language requests are typically command-and-control systems, such as systems for controlling home appliances (e.g. "Switch on the light in the kitchen") or for interacting with devices in a car (Coletti *et al.*, 2003; Moeller *et al.*, 2004). Dialogue-systems that implement such techniques are characterized by the computational model used for the dialogue management. Usually, this computational model is based on a slot-filling paradigm, and often leads to a strongly system-driven style of interaction (S: "Where do you want to go?" U: "To Rome."), well suited for users not very familiar with the system, although it also allows for limited mixed-initiative interaction (U:"Id like to go from Geneva to Rome tonight."), more adequate for users with some a priori knowledge about the system (Aust *et al.*, 1995; Bui and Rajman, 2004). The philosophy behind template matching approaches is that if the system is not able to handle complex tasks, it is not worthwhile to apply sophisticated NLU techniques to process the corresponding potentially complex natural language inputs (McTear, 2002). It is important to understand that template matching does not scale up well to tasks requiring more complex inputs (Milward, 2000; Allen *et al.*, 2001b). For instance, it runs into difficulties when there is negation involved (S:"Which city do you want to depart from?" U:"Lets see, not Geneva") or when several instances of the same type of concept are matched (U: "From Geneva at three or from Nyon at five"). In the first example the problem is caused by the fact that the pattern for a given template does not take into account the context of the match. In the second case, due to the attribute-value formalism used to represent the meaning, the information about which departure city relates to what departure time is lost. As a consequence, it is now quite largely accepted that in domains where complex natural language input is required, the standard template matching approach is not sufficient.

2.3.2 Knowledge-driven understanding with logical forms

When sophisticated language understanding is considered, there is a distinction between the general linguistic *meaning* of a sentence (semantics), and the contextual *interpretation* of the sentence when used in a given situation (pragmatics). The general linguistic meaning can be produced directly from the syntactic structure of a sentence and is often represented as a *logical form* (LF). For example:

$$\text{What states border Texas?} \rightarrow \lambda\text{x.state(x)}\ \Lambda\ \text{borders (x, texas)}$$

Feature-based grammars (using a unification based paradigm) (Jurafsky and Martin, 2009) are a popular resource for this task, as they deal with both the syntactic and semantic features of the constituents that build up a sentence. A contextual interpretation is then obtained by mapping the logical form to a knowledge base representing the available knowledge about the domain, for instance with some variations of first order predicate calculus (FOPC) (Blackburn and Bos, 2003). As opposed to template matching approaches, deep-linguistic NLU has the potential to scale up to tasks of arbitrary complexity. The difficulty lies in the development of the feature-based grammars and the mapping procedures between LFs and the available knowledge base, as well as in the reasoning capabilities required to process complex logical representations (Zue *et al.*, 2000).

To the best of our knowledge, no commercial system integrates a full-blown deep-linguistic NLU module. Laboratory prototypes are extremely limited in the range of tasks and in the size of vocabulary and grammars they deploy (McTear, 2002). However, it is quite widely recognized that research focusing on advanced conversational systems, such as problem-solving assistants (Allen *et al.*, 2001a), reliable natural language interfaces to databases (Popescu *et al.*, 2003), restricted-domain question answering, (Prager, 2006), machine-translation (Rayner *et al.*, 2004), or any type of systems trying to approach human performance in language understanding, is highly dependent on sophisticated NLU. Computational semantics (Traum, 2003; Stone, 2004; Bos, 2005) is a field addressing the hard problem of applying formal semantic theories to computationally tractable models of language understanding. However, achieving practical applications with low manual labor and competence is considered as almost impossible (Glass and Weinstein, 2001). An interesting direction is to take a step back from this type of very ambitious approaches and to make the assumption that practical human-computer interaction does not necessarily have to imply that the system is able to achieve human-like performance in natural language understanding. When keeping in mind that computers are essentially facilitating tools, humans may find more efficient and intelligent ways of interacting without using the full power and complexity of natural language.

2.3.3 Grammatical relations

The idea underlying the approach of using grammatical relations for NLU is that a parser can produce an output that abstracts away the details of the actual sentence but preserves the structures important for understanding. This structure has the form of a set of *grammatical relations* or *grammatical dependencies* (Bunt *et al.*, 2004). For an example, see figure 2.1.

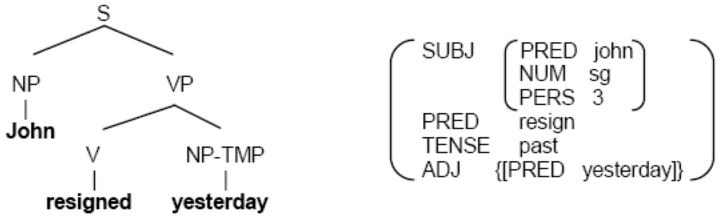


Figure 2.1: Syntactic analysis of “John resigned yesterday” in two forms, as a syntactic tree and as a set of grammatical relations

The grammatical relations are relations like subject (SUBJ), objects (OBJ), indirect object (IOBJ), and relations based on prepositional phrases. Producing such simple relations can be achieved by augmenting, for example, a context-free grammar (Carroll and Briscoe, 2002; Watson *et al.*, 2005). The semantic interpreter may then be a separate process that produces a meaning representation using the grammatical relations as input.

Approaches based on grammatical relations are attractive because the grammatical representations provide a convenient interface between the syntactic processing and complex semantic interpretation procedures, allowing the latter to operate without having to take into account the linguistic complexity of the input. Currently the most important application of grammatical relations is for large-scale language processing tasks, where full parsing provides too much detail and is not robust enough, for instance annotation of text- and spoken-language corpora (Sagae *et al.*, 2003; Watson *et al.*, 2005). However, grammatical relations can also be exploited for the interpretation of natural language input in interactive applications. The advantage of grammatical relations is that they allow interpretations of any complexity. The mapping process can for instance involve inference and discourse processing. In the most complex case, it can be an alternative to the approach based on logical forms (Allen, 1994). In this perspective, approaches to NLU that integrate an efficient interfacing between grammatical relations and semantic interpretation represent a promising direction for the design of flexible natural language understanding engines that adapt the interpretation technique to the complexity of the input.

In this thesis, we elaborate on the use grammatical relations for interpreting questions about recorded, annotated meetings. Meeting data represent a very restricted domain, and the domain-model precisely defines the semantic scope of natural language questions that can be understood by the system. Syntactically, the domain-model does not impose any constraints on what surface structures questions can have. On the other hand, linguistically, the possible grammatical relations that can occur in natural language sentences are limited. Therefore, grammatical relations are an extremely convenient abstraction of syntactic structure, and represent a useful interface between the linguistic domain-independent analysis of questions and the domain-specific semantic interpretation of them. In our implementation, the semantic interpretation is performed in two stages. First a simple concept-spotting technique similar to the one described in 2.3.1 is used to extract instances of domain concepts from natural language questions. Then the grammatical relations are exploited using mapping rules that assign domain specific meanings to pairs of concepts that instantiate a given grammatical relation. The NLU technique is described in detail in 3.5.

2.4 Evaluating natural language interfaces

2.4.1 Technical evaluation

Natural language interfaces are often based on complex implementations of modules that operate sequentially. For example, in a question answering system, some of the natural language processing steps are named entity recognition, word sense disambiguation, syntactic alternation, and logic form transformation (Prager, 2006). In order to evaluate the performance of a natural language interface, both component evaluation and end-to-end system evaluation is important (Jurafsky and Martin, 2009). The component evaluation is performed on specialized data sets that represent the type of input that the component should be able to handle, and reveals weaknesses in the individual NLP tasks. These weaknesses need to be identified because errors made by one component affect the performance of the subsequent components. The end-to-end system is typically evaluated on data sets from real-world contexts, to determine how well the system performs on real tasks. Conferences such as TREC[5] and TAC[6] are examples of evaluation forums that allow for ranking state of the art prototypes based on their performance.

Although the above evaluation methods are an important part of the development of natural language interfaces, they are not sufficient to determine if the system is appropriate for real-world tasks. They do not account for the issue of usability, i.e. how well a real user performs when using the system, and how satisfied they are with the natural language interface. To measure user performance and satisfaction, evaluation methods focusing particularly on usability are needed.

2.4.2 Usability evaluation

The goal of usability evaluation is to determine at least three aspects of a user interface: 1) if the interface supports the user to do their tasks, 2) if the design of the interface makes it difficult or easy to solve the task, and 3) what the user likes and dislikes, and what their understanding of the interface is (Shneiderman and Plaisant, 2009). In this computer-era, where GUIs are the dominating type of user interfaces, it has been recognized that the usability of natural language interfaces is a key factor for making them accepted alongside GUIs (Dybkjaer *et al.*, 2004). NLP components are prone to errors that do not occur with GUIs. For example, when a user clicks with a mouse, it responds appropriately every time, whereas a natural language interface may interpret request correctly in most of the cases, but once in a while misinterprets and whisk the user to an apparently random location. The frustration that rises from the unreliability of NLP can only be overcome if the interface is designed in such a way that users expectations are in line with the system capabilities.

Existing methods for usability evaluation are mainly designed for GUIs. In think-aloud testing the experimenter is present during the evaluation and the subject expresses thoughts and opinions on the system while executing predefined tasks (Norgaard and Hornbaek, 2006). In remote testing the user typically accesses the system online from their personal computer, and the experimenter does not directly observe the user, but the interactions with the system can be logged (Ingensand and Golay, 2009). When an evaluation session has been finished, users are asked about their experiences and expectations of using

[5] http://trec.nist.gov/
[6] http://www.nist.gov/tac/

the system, either in an interview if the study is qualitative, or in a questionnaire, if the study is quantitative (Nielsen, 1995). The information that the evaluator gathers from such evaluations, helps them to improve the visual design of the interface and modify the logic of the interaction when needed.

The standard usability evaluation methods are not straight-forward to apply to natural language interfaces. Think-aloud testing imposes a practical problem. If the interface accepts speech input, the system is not able to distinguish when the user speaks to the system and when he speaks to the experimenter. Also when the interaction is based on keyboard input, talking with the experimenter is likely to interfere in an undesirable way. Remote testing is useful when a telephony-application is being evaluated. However, for multimodal natural language and direct-manipulation interfaces it is more challenging. The user cannot be recorded, and important aspects of synchronized use of modalities may not be possible to log and consequently get lost. Moreover, the system has to be in a relatively final state to be evaluated remotely. The vocabulary and grammars in the NLP components need to cover the type of input that users provide. But one of the objectives of evaluating the usability of natural language interfaces is to gain knowledge about what type of input users want to provide. To overcome this two-way problem, special evaluation methods for natural language interfaces have been created to enable evaluation of unfinished prototypes. They are commonly referred to as Wizard of Oz experiments.

2.4.3 Wizard of Oz evaluation

Wizard of Oz (WOz) evaluations enable users to test a natural language interface at early stages of the system development, before all the NLP components have been implemented. The goal is not so much to find weaknesses or errors in a given interface design, but more to discover how users want to use the interface beyond those ways in which the designers have originally anticipated. In cases where natural language components are designed based on models of human-human communication, early evaluations are particularly relevant, as users may not interact based on human-human protocols. Humans are very flexible in their way of using language. They naturally adapt to their conversational partner (Oviatt *et al.*, 2004). For instance an adult speaks differently with a child than with another adult. In the same way, people adapt their language when speaking with computers (Baber and Stammers, 1989; Bickmore, 2004). For instance, when users believe that the system is unsophisticated and restricted in capability, they adapt their language to match the systems language (Pearson *et al.*, 2006). Since there are many complex phenomena in human-human interaction that may not occur in interaction with a computer, human-human models may even be misleading (Dahlback *et al.*, 1993; Rosenfeld *et al.*, 2001).

On the other hand, what it means to build a natural language system based on models of human-computer interaction is not obvious. To develop a new system based on the natural language input of existing systems is not the optimal way, because current systems are often very naïve and impose users to restrict to a language compatible with the systems limited understanding capabilities instead of using a language that the future system is targeted for. It is therefore not evident that the data produced from existing systems is really useful for the design of more advanced ones. To obtain data that corresponds to what the future system should handle, Wizard of Oz experiments represent the most appropriate evaluation method (Cheng *et al.*, 2004).

In a WOz experiment, the user believes to be interacting with a fully automated system, which, in fact, is controlled by a wizard, who simulates one or several components of the system, typically involving speech recognition, natural language understanding or dialogue management.

There are two distinct purposes for performing WOz experiments. One is the mostly theoretical purpose to aim at characterizing human-computer interaction features in comparison to human-human interaction. The other, more practical, purpose is to provide the empirical data for the development of advanced (not yet implemented) systems. For instance, WOz experiments can be useful for collecting data for speech recognizer training, system requirement specification and for receiving early feedback on a specific dialogue model (Benzmller *et al.*, 2003; Bernsen *et al.*, 2006). WOz experiments can also assist in the evaluation of critical performance tradeoffs and in making decisions about alternative design choices (Oviatt, 2003). The goal is to first explore the design space rather than to try to develop a specific design idea in detail before knowing if it will be relevant in real life situations (Klemmer *et al.*, 2000). The integration of real users early in the development of natural language interfaces has gained importance in recent years, as can be observed from a number of research projects that apply the Wizard of Oz method (e.g. Bernsen and Dybkjaer (2004); Wiren *et al.* (2007); Lee and Billinghurst (2008). Rapid prototyping tools for designing dialogue-systems have started to integrate WOz studies as a central part of their prototyping methodology (Klemmer *et al.*, 2000; Bui *et al.*, 2004; Cenek *et al.*, 2005).

Although the Wizard of Oz method has become a standard for evaluating natural language interfaces, it has mainly been used for voice-only applications, not multimodal language and direct manipulation interfaces. Currently there are few guidelines and experience reports available on how to extend this relatively complex evaluation method to language-enabled GUIs. Within the framework of this thesis, we have developed an experimental setup for Wizard of Oz evaluation of multimodal natural language and direct manipulation interfaces to meeting data, and we contribute to the state of the art by identifying important issues in the design of the wizards control interfaces. More detail can be found in 4.3.

3 Natural language querying of meeting discussions

This chapter addresses a particular case of information retrieval, namely the retrieval of answers to questions about spontaneous, spoken meeting discussions. Two important questions are addressed. First, can meeting discussions be enhanced with structural information so that a precise episode in which a given event or argumentation occurred can be retrieved? Second, can a natural language understanding module be built that correctly interprets questions about these events and argumentations? To determine the importance of adding argumentative annotation on discussions, we analyze a collection of questions about past meetings. We show that topical and argumentative annotation of meeting discussions are indeed essential for answering questions in this domain. We also make an annotation study with multiple annotators on a set of transcribed meetings, to determine how difficult it is to recognize argumentative categories in real, recorded discussions. Here we show that real discussions are highly ambiguous with respect to formal models of argumentation, and that individual argumentative contributions need to be classified with multiple argumentative categories in order to account for different questions that can refer to the same episode of discussion. For the understanding of questions we implement a natural understanding module that works in two steps: a domain-independent NLP component that generates a logical form of the question, followed by a domain-specific semantic component that extracts attribute-value pairs according to a domain model of meeting discussions. With this implementation, we demonstrate that a relatively small set of generic rules is sufficient to interpret a wide range of syntactically and lexically heterogeneous questions. The key requirement is to assign syntactic roles to words and phrases during the linguistic analysis phase. We conclude that efficient meeting data retrieval depends on at least two factors - enhancing the data with appropriate structural information, and using linguistic tools to interpret questions. If any of the two are neglected, a large fraction of questions risk either to not get the correct answer retrieved or to get incorrectly interpreted.

3.1 Introduction

In recent years there has been an increasing interest in research on developing systems for efficient access to multimedia meeting data. In work situations where meetings are a crucial part of a team's project progress, it is often important to be able to refer back to the agenda, discussions and outcomes of past meetings. Currently there are no standard information systems for this need. The data, when available, is heterogeneous and scattered across various media. E-mails are sent to announce upcoming meetings and meeting agendas. Documents are presented during meetings. Notes are written on whiteboards or by hand. Minutes are produced after the meeting. The meeting discussion itself represents an important source of information that is rarely captured as such.

The challenge in developing meeting data retrieval systems lies in gathering all this data into a database and making it searchable. From a technology point of view, the first problem is already solved. Meetings can be captured in Smart Meeting rooms (Nijholt *et al.*, 2006) where participants have individual microphones to record their speech into different audio channels. Whiteboards are electronic, and notes made on paper are written with pens that capture the writing electronically. Powerpoint presentations are synchronized with the meeting recording, so keep track on which point in time a given slide was presented.

The second problem, i.e. making the data searchable, is a more open research question. To enable search in meeting discussions, basic post-processing of the meeting recordings is advisable, typically (automated) transcription of meeting discussions, speaker segmentation, and utterance segmentation. More advanced processing involves enhancing the data with topic segmentation and labelling, and discourse segmentation (dialogue acts and argumentative annotation). Such higher-level annotations can be done in many different ways, using different annotation schemes and guidelines. For example topic labels can be generated by extracting the most representative keywords from the discussion, or by selecting a more general semantic concept to describe the episode.

Once the data has been enhanced with annotations to facilitate search, there is the question of user interface to access that data. One of the main goals of the user interface is to succeed in getting answers to questions such as "What decisions were made?" and "Why was a given proposal turned down?" In this type of information retrieval where the data is spontaneous, spoken conversations, the content itself represents only part of the information that is present in the data. Standard keyword-based document retrieval is therefore too limited for this type of application. Another, possibly more important, part is the argumentation behind the various outcomes. To express queries about the argumentation, we believe that natural language represents an intuitive search modality. Hence, we propose a user interface to meeting data that operates as a natural language query engine. The user can ask questions freely in natural language, and the system retrieves the meeting episodes that provide the answers to those questions.

When developing a natural language search interface to natural language data there are two sides to the problem, both of which are addressed in this chapter:

1. Meetings need to be annotated in such a way that users can retrieve answers to questions in an intelligent and efficient manner. The more fine-

grained the annotation is, the more different types of questions can be answered.

2. The natural language understanding in the system needs to be able to interpret questions that refer to the argumentative process in the discussion. The *fewer and more generic* the natural language interpretation rules are, the more robust the system will be in providing interpretations to different questions

In this chapter, we first analyze a set of natural language queries on meeting data, collected as part of a user requirement analysis for a future meeting data retrieval system. We show that questions about argumentation are so frequent that they motivate the effort of annotating the argumentation in meeting discussions (3.2). We then specify the general system architecture of a natural language query engine to meeting data, with the required system components for retrieving answers to complex queries about meeting discussions (3.3). To assess the difficulty of annotating the argumentations in discussions, a hands-on annotation study was made with five annotators on three meetings, using an argumentation schema proposed by Pallotta and Ghorbel (2003). The study revealed that some argumentative actions are highly ambiguous, such as the proposal of a new issue (3.4). The same argumentation schema was used for natural language understanding (NLU) of questions. We developed a small-scale NLU module based on 55 interpretation rules that matched syntactic and semantic elements in the natural language query with concepts in a predefined meeting domain model (3.5). The conclusion of this work was that few rules are sufficient to account for a large variety of questions, given that linguistic resources are used for parsing the question and semantic roles are assigned to its constituents (3.6)

3.2 Question types

Natural language querying of meeting discussions is a special case of both information retrieval (IR) and question answering (QA) and at the same time differs in crucial ways from both fields. It is not standard IR, because question in this domain can rarely be answered by matching terms in the query with keywords in the discussion. It is also not standard QA, because the answer to a question does not have the same syntactic and lexical properties as the question. The question types considered in TREC-QA evaluations (Voorhees and Buckland, 2007) are limited to definition, factoid and list questions, all of which are not representative of questions on meeting discussions.

There has been work on spoken language IR that has focused on limited types of questions related to meeting data retrieval, namely topic-based and dialogue-act based questions (Vinciarelli, 2005; Stolcke *et al.*, 2000; Clark and Popescu-Belis, 2004). However, meting data can have many more dimensions that users may refer to in questions, such as the argumentative process and outcomes of a discussion, the meeting agenda, and referenced documents. A useful natural language query engine to meeting data should include at least some of these dimensions in order to meet the demands of real-world users. But choosing which dimensions to invest effort on is not obvious. The real-world need has to be investigated first.

In the early stages of system development, a common practice is to perform a formal user requirement analysis to gather realistic user needs for a given task. These user requirements then set the ground for the design of the future system. In the context of meeting data retrieval systems, several user requirement studies have been made in the past to find out what type of information users want to access in meeting data, in order to design systems with the adequate search functionalities to meet these needs (Lisowska *et al.*, 2004; Banerjee *et al.*, 2005; Cremers *et al.*, 2005). The user requirement analysis performed by Lisowska *et al.* (2004) is of particular interest to us. It was done as part of the Swiss Interactive Multimodal Information Management (IM2) project[1], which also sets the framework for the work in this chapter. The analysis was performed as a questionnaire survey in which subjects were asked to imagine themselves in one of four use case scenarios, and based on the chosen scenario write down questions that they would ask to the system. The four use case scenarios in the survey were:

- an employee who had missed a meeting on a project they are involved in and wants to catch up
- a new employee who is using the system to familiarize themselves with a project that they will be involved in
- a manager who is tracking the progress of a project
- a manager who is tracking employee performance

The majority of the survey participants chose the first or the second scenario. Hence, most of the queries in the set represent requirements for users who want to catch up on missed meetings.

One of the important findings was that the queries could be divided into two broad categories: 1) queries that pertain to elements related to the interaction among participants, e.g. agreement/disagreement, proposals, argumentation (for and against), and 2) queries that pertain to concepts in the meeting domain, e.g. dates, times, participants, and topics. We refer to them as argumentative (about the argumentative process and outcome of the meeting), and non-argumentative or factual (about the meeting as a physical event, or thematic). In this section we want to assess the difficulty of answering questions in terms of the knowledge and inference capabilities required in the system, in particular the need for annotating the argumentation in meeting discussions to answer questions. The above classification does not provide sufficient insight. In particular, it does not take into account questions that pertain to a mix of different types of information. Therefore, we complement the analysis with the following categorization:

- queries that can be answered using standard IR techniques on meeting artefacts only, e.g. minutes, written agenda

[1] http://www.im2.ch/

- queries that can be answered with IR on meeting recordings
- queries for which IR does not apply or is insufficient and for which additional information and inference is required, e.g. about the meeting participants, meeting dynamics, external information about the projects discussed in the meetings

It is important to note that query elicitation through survey studies can be criticized for providing a biased data set. For example, if the participants in the survey are part of the same project in a company, their queries tend to be homogenous. To ensure that we have a maximally heterogeneous, unbiased and realistic data set, we use three different datasets for our analysis:

- The IM2 dataset by (Lisowska *et al.*, 2004) with 270 introspective questions that are not related to any particular recorded meeting
- The BET observations (Wellner *et al.*, 2005), a set of 294 natural language statements about existing meeting records, elicited in a study where subjects were asked to watch a meeting recording and report observations of interest for the participants in the meeting. This data-set is used as cross-validation of the IM2 set. An IM2-query is considered as 'realistic' if there is a BET observation that represents a possible answer to the query. For example, the query "Why was the proposal made by X not accepted?" matches the BET observation "Denis eliminated Silence of the Lambs because it was too violent".
- The Manager Survey, a new small set of queries that we collected in a survey addressed at managers of companies. This set consists of 35 queries.

The queries were analyzed by two teams of judges. The first part, which consisted in cross-validating the IM2 set with the BET observations, gave 90 queries that were judged as being valid, and hence representing the most realistic questions in the IM2 set. Next, each team analyzed the IM2 set and the Manager Survey set by discussing each query individually, and classifying it according to query type (factual, thematic, process and outcome), and query difficulty. The full details are reported in Pallotta *et al.* (2007). Here we give an account of the second dimension, query difficulty.

Query difficulty was assessed by assigning queries to one or several of 10 categories, according to the type of information and techniques judged as necessary for answering the query. The 10 categories were:

1. *Role of IR*: the relevance of standard[2] Information Retrieval and topic extraction techniques for answering the query. The possible values are:
   (a) *Irrelevant*: IR techniques are not applicable. Example: *What decisions have been made?*
   (b) *Successful*: IR techniques are sufficient. Example: *Was the budget approved?*

[2] By standard IR we mean techniques based on bag-of-word search and TF-IDF indexing

(c) *Insufficient*: IR techniques are necessary but not sufficient alone. Additional information such as argumentative, cross-meeting, external corporate/project knowledge, or inference is required. Example: *Who rejected John's proposal about the layout of the room?*

2. *Artefacts*: information such as agenda, minutes of previous meetings, e-mails, invitations and other documents related and available before the meeting. Example: *Who was invited to the meeting?*
3. *Recordings*: audio, video or transcription of the meeting. This information is needed to answer most questions. Example: *What did Mary present?*
4. Metadata: contextual, static knowledge about the meeting and its participants. Example: *Who were the participants at the meeting?*
5. *Dialogue acts and adjacency pairs*: Example: *What was John's response to my comment on the last meeting?*
6. *Argumentation*: annotation of the argumentative structure of the meeting content. Example: *Did everyone agree on the decision, or were there differences of opinion?*
7. *Semantics*: semantic interpretation and reference solution of terms in the query. Example: *What decisions got made easily*?
8. *Inference*: deriving implicit information, calculation, and aggregation. Example: *What would be required from me*?
9. *Multiple meetings*: cross-meeting information. Example: *Who usually attends the project meetings*?
10. *External*: knowledge related to the project or corporation and not explicit in the meeting discussion. Example. *Did someone talk about my work*?

The role of IR techniques in answering queries is presented in table 3.1. The results are given both for the query sets as whole (IM2-set and Manager Survey-set), and for the subset of each query set that pertains to the argumentative process and outcome of meetings. We found that a strikingly low number of queries can be answered successfully with IR and topic extraction alone (IM2: 14%, MS: 20%). For the remaining queries, IR is either insufficient (MS:54%) or irrelevant (IM2:50%). If we consider only argumentative queries, the numbers are even more extreme. IR techniques are never sufficient to answer them.

The additional information and retrieval techniques required to answer queries when IR alone fails, are shown in table 3.2. When read column-wise it shows the frequency of different combinations of information and retrieval technique categories (categories 2-10) to answer a query. For example, the most frequent combination in the IM2 set is meeting recording, argumentative annotation, semantics and inference. When read row-wise, the table shows how often individual information categories occur in the combinations. Artefacts, dialogue acts and adjacency pairs, multiple meetings and external knowledge are relevant for answering only a very small fraction of queries. Meeting recordings enhanced with argumentative annotation on the other hand reoccur in 8 of the 12 most

IR is:	**IM2 set (270 queries)**		**MS-set (35 queries)**	
	All queries	**Argumentative**	**All queries**	**Argumentative**
Successful	14.4% (39)	0.8% (1)	20.0% (7)	5.3% (1)
Insufficient	35.6% (96)	52.1% (61)	54.3% (19)	78.9% (15)
Irrelevant	50.0% (135)	47.0% (55)	25.7% (9)	15.8% (3)

Table 3.1: The role of IR and topic extraction in answering users' queries.

frequent combinations. This means that argumentative annotation is a very important dimension of meeting data and needed for answering the majority of questions in this domain.

	IM2-set											
Artefact				x								
Recording	x	x	x	x	x	x	x	x		x		x
Metadata			x			x		x	x		x	
Dlgacts/Adj.pairs												
Argumentation	x	x	x	x	x			x		x		x
Semantics	x	x	x	x	x	x	x	x	x			
Inference	x		x	x	x	x	x		x			x
Multiple meetings					x		x		x			
External												
Cases	29	11	9	9	8	8	8	7	7	5	5	4
Ratio (%)	12.5	4.7	3.8	3.8	3.5	3.5	3.5	3.0	3.0	2.1	2.1	1.7

Table 3.2: The most frequent combinations of information required for answering queries in the IM2-set

Overall, our user requirement analysis indicates that topical, argumentation and semantics are the key categories of information that need to be considered when developing a natural language query interface to meeting data. Inference is also important, but in practical applications it is the most difficult to realize. A reasonable trade-off in this case is to find retrieval techniques that point out the relevant episodes of meeting discussion from which the user can then infer the answer to their potentially complex query. We argue that inference is a skill that humans master with excellence and without effort, and that this should be used as an advantage in the design of the system rather than developing artificial inference techniques. From this perspective, the meeting data retrieval system that we design in this chapter is grounded on the following fundamental requirements which we derived as important from our query analysis:

- The central piece of data to be retrieved is meeting recordings, either as video or transcript.

- Answers to questions are to be inferred by the user by watching or reading the relevant piece of meeting episode.
- Meetings are to be annotated both with topical and argumentative structures to find the relevant episode of meeting discussion.
- Queries are to be interpreted with linguistic tools. Syntactic analysis, followed by domain-specific semantic interpretation of terms is required.

In the next section we describe the general architecture of the natural language query engine.

3.3 Architecture of a natural language query engine to meeting data

The architecture that we propose for retrieving meeting data with natural language queries consists of four main system components, or five, depending on the technique chosen for the first component, the automated speech recognition (ASR). ASR is not in the scope of this thesis, but we briefly describe the available techniques in order to motivate the architecture chosen for this particular application.

There are two main approaches to ASR: the statistical and the grammar-based. Both have advantages and disadvantages with regard to accuracy and robustness of the natural language processing as whole. The statistical approach uses non-linguistic probabilistic models to generate a word lattice, i.e. a compact representation of different hypotheses of what the user said. The advantage of this approach is that every speech input gives a result, and the ASR does not commit to producing one specific string of words, but lets the following NLP components (syntactic or semantic analyzer) disambiguate the word lattice and select the most probable sequence of words. The approach is very robust, because the syntactic and semantic analyzers have a large choice of possible input strings, and there is a high chance that they can find a word sequence in the word lattice that can be parsed by the grammar and receives a semantic interpretation in the domain. The disadvantage is that the ASR uses domain-independent resources for processing the speech signal and cannot be trained to recognize typical sentences in the given application, hence not guaranteed to yield accurate hypotheses for the syntactic and semantic analyzers to work with.

The grammar-based approach has the opposite characteristics. Instead of processing the speech input in sequence (ASR, syntactic analysis, semantic analysis), the ASR uses the syntactic and semantic resources as application-specific knowledge to produce the most probable string of words. As a bi-product of using linguistic resources it generates the syntactic-semantic analysis of the sentence, ready to be processed by the next system component. Here the main advantage is that the string can be generated with very high accuracy if the syntactic and semantic resources are tuned to reflect typical sentences in the application domain. The disadvantage is the lack of robustness. Unforeseen speech input that does not match the available vocabulary or syntactic structure in the linguistic resources receives no interpretation at all.

In the current design of the natural language query engine we choose grammar-based ASR for the accuracy in disambiguating and generating a linguistic analysis of the query. The complete system architecture can be viewed in figure 3.1. System components and resources that are of particular interest to the current research are marked in bold. A query is processed sequentially by four system components. The first is the grammar-based speech recognition which generates a syntactic-semantic representation of the query, called 'logical form'. We describe it in more detail in section 3.5. The next component is the domain-specific natural language understanding (NLU) component that interprets the logical form and maps it to concepts and relations in the meeting domain. The output is a set of attribute-value pairs called 'semantic constraints', also described in more detail in 3.5. This set is then processed by the database query generator into a form that matches the chosen database technology, in this case into an SQL-query, as the database is relational. The last component executes the SQL-query on the meeting database and retrieves the meeting episode(s) that contain the answer to the query. In this chapter, the main contributions relate to the development of the meeting database and the NLU component.

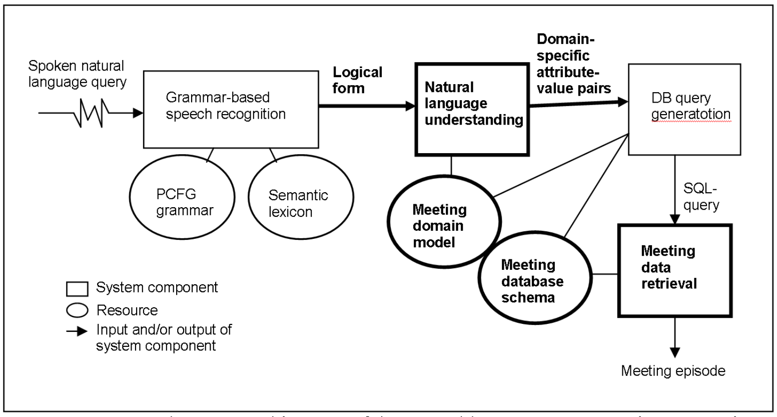


Figure 3.1: General system architecture of the natural language query engine to meeting data

The database is the central element of the system, storing meeting data in multiple data layers. The database schema is represented in figure 3.2. The most fundamental layer is the transcription of the utterances in the meeting discussion and their corresponding dialogue acts. To this layer is added the higher-level annotations that were made on meeting segments. One such layer is the topical or thematic segmentation and labeling of meeting episodes, another is the argumentative structuring. There is also a layer for documents that were referred to in utterances, and one for speakers who produced the utterances. Finally, a metadata layer is included, containing information about the places, dates and participants of meetings. The database schema was developed progressively by many partners. An early version was reported by (Armstrong *et al.*, 2003), and later improved by (Melichar, 2008). This work contributes to the argumentative layer of the database by evaluating an argumentative annotation schema under development and populating a meeting database with these annotations.

The approach taken to natural language understanding of questions heavily relies on the structure of the database and its data layers. Questions are interpreted based on their references to attributes and values in the meeting domain. However, for general applicability, the user queries should not rely on a specific database schema or technology behind it. Their interpretation should be independent of specific types of data modeling. To this end, our NLU component uses a meeting domain model that describes the domain in three types of elements only: concepts, attributes and relations (see figure 3.3).

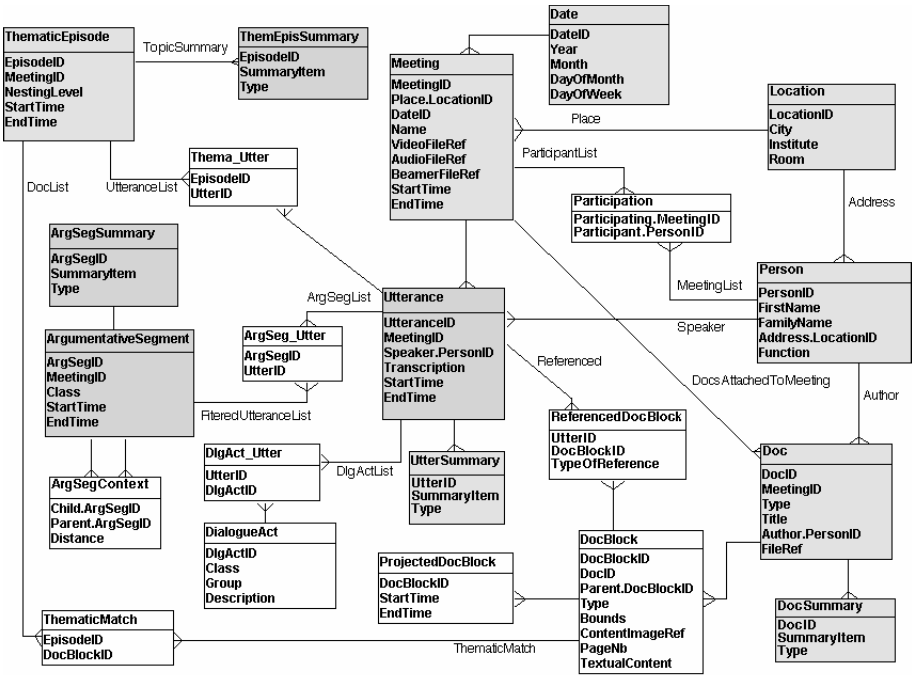


Figure 3.2: The meeting database schema

The interpretation of the query is represented as a database-independent canonical form using these concepts, attributes and relations. For example the query *"What suggestions did John make about the layout of the room?"* could give the following interpretation:

Syntax of canonical form	**Interpretation of query**
ConceptInstance.Attribute = Value	person1.firstname=John
	argseg1.cat=suggestion
	topic1.label='layout of the room'

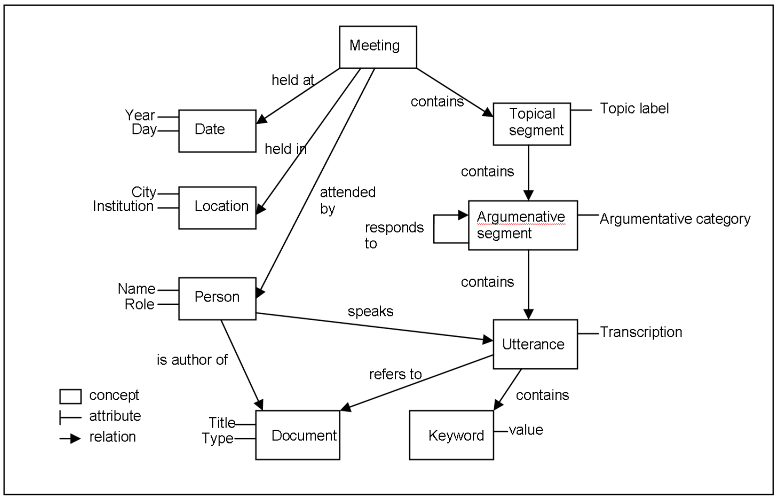


Figure 3.3: The meeting domain model

Relation(Concept1, Concept2)	speaks(person1, argseg1)
	contains(topic1,argseg1)

In the remained of the chapter we describe the practical work that was done on argumentative annotation of meeting discussions and the development of the NLU component. Experimental results are presented and discussed.

3.4 Argumentative annotation of meeting discussions

3.4.1 Argumentation structures in meeting discussions

Modeling human argumentation is a task that has many potential applications, for example argument summarization (Delannoy, 1999), computer-supported collaborative argumentation (CSCA) (Gordon and Karacapilidis 1999), and enhanced meeting data retrieval (Pallotta and Ghorbel, 2003). The approach to dialogue modeling largely depends on the targeted application. For example, in CSCA the goal is to model the 'ideal' argumentation structure that will enforce participants of a live meeting to argue in a more efficient manner, whereas in meeting data retrieval the goal is to model argumentation as it occurs in spontaneous conversation, in order to be able to annotate the structure of recorded conversations and make them searchable. In the CSCA application the argumentation model is prescriptive and static, whereas in the data retrieval application it is descriptive and should be flexible to real-world phenomena.

In this work, the goal is to use an argumentation schema to annotate transcribed meeting discussions in order to enable complex natural language querying of the data. The requirement on the argumentation schema is that it contains the types of categories and relations that users would refer to in real-world queries. In section 3.2 we showed that queries in the meeting domain typically refer to the topical and argumentative layers of meeting discussions. More specifically, the argumentative terms that frequently occur in queries are: ”suggest”, ”decision”, ”objections”, ”turned down”, ”consensus” etc.

The argumentation schema proposed by Pallotta and Ghorbel (2003), based on the IBIS model of argumentation (Kunz and Rittel, 1970) answers the current need. In this schema argumentation is structured hierarchically, as shown in figure 3.4. In the shown example, the highest-level segment has the category Discuss, and smaller segments within this segment have the categories Propose, Accept and Reject. Further, each argumentative action is described with a specific role that the action has in the discussion. For example, when a proposal is made, the role of the proposal can be to provide a new idea, a solution to a problem, or an alternative to a previous proposal. Notice that the hierarchy of the schema is not intended to be used to model a whole meeting discussion as a tree. The Discuss-category is the top-category of the hierarchy, but there can be any number of Discuss-segments in a meeting. Also, the schema is not intended to be used for annotating all utterances in a meeting with argumentative categories. There can be episodes in meetings that are not argumentative at all and therefore should not be labelled as such.

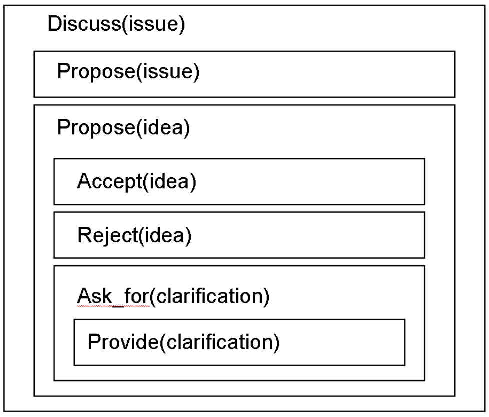


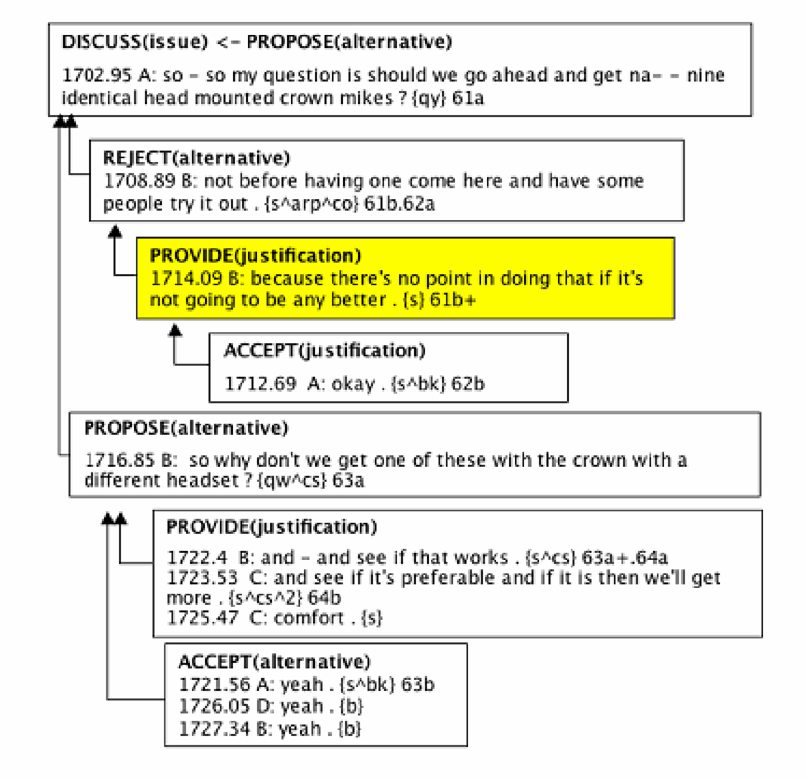
Figure 3.4: Hierarchical representation of argumentative structure

The main goal of the hierarchical structure is to enable retrieval of answers to questions that refer to complex structures in discussions, for example:

”Why did John reject Mary’s proposal?”.

For the above example, the retrieval task consists in identifying three argumentative segments in a hierarchy, namely Mary’s proposal, John’s rejection, and his justification to the rejection. Figure 3.5 shows an excerpt of a meeting that was annotated with the proposed hierarchical structure. The highlighted segment represents a potential answer to the question.

It is worth mentioning that Pallotta's and Ghorbel's argumentation schema is relatively ambitious in its complexity compared to other existing argumentative schemas designed for meeting data retrieval e.g. Galley and Mckeown (2003); Hillard *et al.* (2003). The contrasts between models can be explained by the different objectives in developing annotation schemas for this application. The first is to maximize querying capacity, which is the focus of this chapter and motivates the choice of a complex argumentation schema. The second is the ability to annotate meetings automatically, possibly on *automatically* transcribed data, which motivates the choice of a more simple schema. This second objective is at least as important as the first. In order to employ meeting information systems into real-world usage, companies that use such systems need to be able to update the database continuously. Manual transcription and annotation is in this context expensive and unrealistic. To ensure high performance on automated annotation, simplicity of the argumentation schema is a key factor.



DISCUSS(issue) <- PROPOSE(alternative)
1702.95 A: so - so my question is should we go ahead and get na- - nine identical head mounted crown mikes ? {qy} 61a

REJECT(alternative)
1708.89 B: not before having one come here and have some people try it out . {s^arp^co} 61b.62a

PROVIDE(justification)
1714.09 B: because there's no point in doing that if it's not going to be any better . {s} 61b+

ACCEPT(justification)
1712.69 A: okay . {s^bk} 62b

PROPOSE(alternative)
1716.85 B: so why don't we get one of these with the crown with a different headset ? {qw^cs} 63a

PROVIDE(justification)
1722.4 B: and - and see if that works . {s^cs} 63a+.64a
1723.53 C: and see if it's preferable and if it is then we'll get more . {s^cs^2} 64b
1725.47 C: comfort . {s}

ACCEPT(alternative)
1721.56 A: yeah . {s^bk} 63b
1726.05 D: yeah . {b}
1727.34 B: yeah . {b}

Figure 3.5: Argumentative structuring of the ICSI meeting data (Bmr012)

The long-term goal of this work is to meet also the second objective of annotating meetings automatically. However, our approach is to start from user requirements and first validate the usefulness of a hierarchical argumentation schema for retrieving answers to questions in meeting discussions. When the schema has been shown to be appropriate for the task, simplifications can be made, and the impact of these simplifications on the retrieval task evaluated.

In order to validate the usefulness of a hierarchical argumentation schema for the intended task, real meetings need to be annotated with that schema. First, it allows us to test if the categories of the schema are appropriate with respect to real meeting discussions. Secondly, it provides the data layer for the physical database which can then be used for performing real retrieval tasks. The following sub-sections describe the annotations of three recorded meetings using Pallotta's and Ghorbel's schema.

3.4.2 The annotation task

To validate Pallotta's and Ghorbel's annotation schema on actual recorded meeting discussions, an annotation task was designed for this purpose. The task consisted in annotating three transcribed meetings that revolved around reaching a decision on an issue, i.e. meetings where argumentation was dense. The meetings were taken from different sources and had different topics. The three meetings were:

- A movie club meeting where the goal was to choose a movie for the next screening (MovieClub)
- A meeting about furnishing a reading room in an institution. The goal was to choose which pieces of furniture to buy, what colours to choose and how to place the pieces in the room (Furniture)
- A weekly meeting of an ICSI work team at the international Computer Science Institute in Berkeley. The overall goal of the meetings was to collect a corpus of transcribed meeting recordings.

All meetings were recorded in a Smart Meeting Room. The two first ones were 'simulated', which means that the participants were assigned roles to act upon. There was however no precise manuscript. The discussions were spontaneous and were considered to represent realistic argumentations. The third meeting was 'natural' in the sense that it would have happened anyway, even if it hadn't been recorded in a Smart Meeting Room. When meetings are being recorded, there is an underlying assumption that participants behave slightly less spontaneously and more cautiously than when they are not recorded, hence leading to a somewhat less natural argumentation even though the meeting is real.

Five annotators were involved in annotating different subsets of the three meetings. Each meeting was annotated by at least three annotators. In the annotation team three were considered as experts, and two were novice annotators. The aim was to find out if the annotation task was intuitive both for experts and non-experts.

The annotators were provided with annotation guidelines describing in detail the argumentative categories to be annotated, and the links between them, for example:

Disagree (=reject): A contribution that expresses disagreement with another participant's standpoint. Disagreement might be followed by (or contain in the same turn) Justification of why the speaker disagrees and should be annotated as a Justification which is linked to the Disagree segment by the Elaborates relation.

The challenge of writing annotation guidelines is that the definitions of each category should be as unambiguous as possible so that annotators will not confuse two categories. In practice the goal is that different annotators agree that a given segment is of a given category. From this perspective, the original annotation schema with multiple roles on each category (e.g. Propose(solution), Propose(alternative), Propose(idea)) was considered as too detailed. We decided to exclude the roles for this annotation study. The final annotation guidelines therefore consisted of eight basic categories: Propose, Agree, Disagree, Justify, Explain, Request justification, Request explanation, and Decide. The links between categories were of two types and were defined in the following way:

1. *Replies_to:* links two segments whose turns belong to different speakers (e.g. Agree Replies_to Suggest).
2. *Elaborates:* links two segments whose turns might belong or not to different speakers (e.g. Justify Elaborates Disagree).

Annotators were requested to assign one or more argumentative categories to one or more utterances. In other words, an argumentative segment could consist of any number of utterances. Such an annotation freedom naturally makes the task very difficult. Annotators have to agree not only on the fact that a given contribution is of a given category, but also when the contribution starts and ends. This is particularly ambiguous when a speaker elaborates on their own contribution. For example, how does one recognize the border between when a proposal is finished and the explanation of this proposal starts; or when a disagreement ends and a justification starts? Nevertheless, we wanted to give annotators maximal freedom in order to find out if these boundaries can be identified intuitively.

3.4.3 Experimental results

The three meetings in the annotation study had different length and argumentative density. The difference is particularly evident between the simulated Movie club meeting and the natural ICSI meeting (see table 3.3). The two meetings were of approximately the same length, but in the natural meeting, participants spoke twice as much as in the simulated one. The number of identified argumentative segments, however, was almost equal. This demonstrates the effects of simulation of meetings on the argumentation. Participants express themselves more economically, avoiding social and private communications in the meeting.

When computing the inter-annotator agreement on each meeting, we found that there was a relatively low agreement on the overall task, as can be seen in

Meeting	**Time**	**Utterances**	**# Argumentative segments**
Movie club	49 min	1008	259
Furniture	18 min	686	229
ICSI	46 min	2224	283

Table 3.3: Characteristics of the meetings in the annotation study

table 3.4. This can be explained by the difficulty of deciding both the segmentation and annotation of each category, and the number of categories in the study. The highest agreement was achieved on the ICSI meeting, suggesting that it is easier to recognize argumentative segments in spontaneous natural discussions than simulated ones. There was virtually no agreement on the links between segments. This we believe had more to do with the annotation guidelines and examples that were given in it, and needs to be investigated further. Interestingly, no difference could be observed with regard to the inter-annotator agreement among experts and the agreements across novice and expert annotators.

To understand why the inter-annotator agreement on the overall task was so low, we looked at each category separately to see where the disagreements occurred. The results for two annotators on the ICSI meeting can be seen in table 3.5. The highest agreement was achieved on the two categories Propose and Request justification. Here we observed that contributions in meetings often contained cues that helped to identify the category. In case of Propose, the word choice at the start of the utterance gave the cue, for example: ”I think”, ”May I propose”, and ”I suggest”. In case of Justification request, the utterance was typically a why-question.

Meeting	Kappa score
MovieClub	0.429
Furniture	0.426
ICSI	0.453

Table 3.4: Inter-annotator agreement

The most frequent disagreements between annotators occurred for certain pairs of categories. One such pair was Justification and Propose. The reason why annotators often confused these two categories was that the statements in meetings were in fact ambiguous with respect to these two categories. When a speaker disagreed, they often justified the disagreement by suggesting a better idea. The same utterance could hence have two interpretations: Justification and Propose. We observed the same confusion between Disagreement and Propose.

The second typical confusion occurred between the pair Disagreement and Justification and the pair Propose and Explanation. Here the confusion was due to the difficulty in recognizing the border between the end of the disagreement and the start of the justification, and likewise for the proposal and explanation. In many cases, one annotator marked the whole segment as just a disagreement whereas another one would divide it into two: disagreement followed by justification. Also, in cases where a disagreement and justification were part of the same utterance, some annotators classified it with one category and others with two.

	Propose	Agree	Disagree	Req.Expl.	Req.Just.	Explain	Justify	Decide
Propose	695	14	52	26	19	40	191	72
Agree	34	67	9	2	1	12	17	25
Disagree	37	4	130	5	2	4	35	15
Req.Expl.	34	2	5	86	5	5	16	26
Req.Just.	21	1	0	12	114	13	9	7
Explain	59	24	9	9	10	117	28	4
Justify	127	6	46	15	11	20	320	59
Decide	45	2	12	26	8	3	41	69

Table 3.5: Confusion matrix showing disagreements between two annotators on the eight argumentative categories when annotating the ICSI meeting

The lowest agreement was achieved for decisions. This may surprise some readers, as it seems intuitive to recognize episodes where decisions are made. In reality, decisions are rarely explicit as unique sentences in the meeting discussion, e.g. "Ok, let's decide to take this one." Decisions are often achieved by mutual (silent) agreement after some discussion. If an annotator is forced to select a segment that corresponds to the moment of decision, it may turn out be a very short sentence, e.g. "Ok". The question is if annotating that segment as Decision is helpful for retrieving answers to questions. "Ok" does not represrent an answer to "What decisions were made?". It only shows that a decision was made. It is in fact debatable whether decisions can be annotated as a category at all. The low inter-annotator agreement is therefore justified.

3.4.4 Conclusion

The obtained results indicate that real meeting can be annotated with argumentative categories according to an argumentation schema, but it is not unproblematic. The main problem is that, regardless of schema, the argumentative contributions in meetings will always be ambiguous. Utterances can be interpreted differently depending on what is the main goal of the annotation. We see three approaches for addressing the problem of low inter-annotator agreement:

1. Make the annotation task easier. For example, in cases where it is hard to determine the border between segments that are spoken by the same speaker, as in the case with Propose and Explain, collapse these into one and give the segment two argumentative categories. Although such a structure is simpler than the original hierarchical one, it is still useful for answering questions on the data. The segments are bigger and less precise, but the answer is still contained within the retrieved segment.

2. Reduce the number of categories. For example exclude categories that represent elaborations, such as explanations and justifications. But the simplification of the schema means that questions with subtle differences cannot be distinguished, and receive the same interpretation.
3. Allow segments to be annotated with several categories and ignore inter-annotator disagreement. Disagreements can in fact be positive. It means that different questions can be answered by retrieving the same episode of meeting discussion, which may lead to a more useful retrieval system. For example, the two questions "What did John suggest?" and "Why did John not agree with Mary's proposal?" may be answered by the one and same segment in the meeting, where John proposes to use a different solution than Mary because hers is for example too expensive.

From a question-answering perspective, we conclude that a rich and ambiguous set of argumentative categories gives more chance of retrieving relevant episodes of discussion than a simple, unambiguous set of categories. We also conclude that meeting data retrieval benefits from annotating several categories to the same segment rather than annotating each segment with only one category.

3.5 Prototype of a natural language question understanding engine

3.5.1 Model-driven approach to natural language understanding

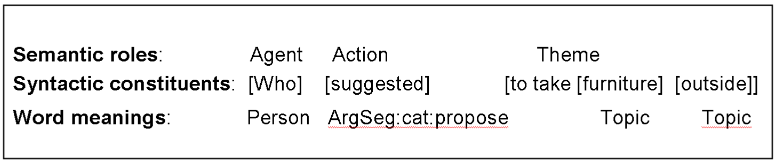
In section 3.3 we described the architecture of a natural language query engine to meeting data where the natural language understanding of queries is based on a domain-model of the meetings. We showed that meeting data have many information layers, including topical, argumentative and metadata layers. The underlying idea of a model-driven approach to natural language understanding is that from the system point of view, it is only worthwhile to "understand" queries that match something in the system's internal world of knowledge, i.e. the database. Even if the system was able to understand queries outside the scope of meetings, it would not be able to answer them. The second motivation for model-driven NLU is that one can expect from the user to cooperate with the system by asking questions that the system is able to answer. Since applications are generally created for specific tasks, the user should be aware, at least globally, of what the system is able to understand and what it is not.

The "understanding" of natural language queries is, in this sense, nothing more than a mapping between the linguistic form of the query and the domain-specific query representation that the system is able to use for performing computational tasks. In our case, the goal is to transform a natural language query into attribute-value pairs, from which a precise SQL query can be formulated and then executed on the multi-layer database. The task at hand has certain similarities with Natural Language Interfaces to Databases (NLIDB), because the ultimate goal is to translate the NL question to a database query. But in this work, we limit ourselves to producing the intermediate, database technology-independent formal representation. A particular focus is given to queries that address topics, speakers and argumentation in meeting discussions.

To illustrate the model-driven approach to natural language understanding, we give an informal example of the different stages of the understanding process. Consider the query:

Who suggested to take furniture outside?

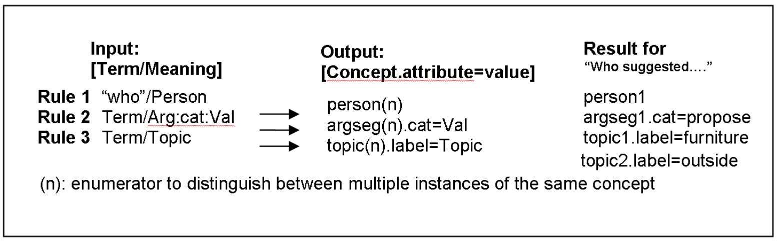
In this query, the terms that refer to concepts in the domain model are underlined. "Who" refers to an instance of Person, "suggested" to Argumentative segment, and "furniture" and "outside" to Topic. Moreover, these instances of concepts have relations. For example, the instance of Person speaks the instance of Argumentative segment. In order to translate the query into appropriate domain-specific attribute value pairs, the query has to be processed in several steps. First, it has to be analyzed lexically, syntactically and semantically using linguistic resources. This processing provides a syntactic segmentation of the constituents, semantic roles of those constituents, and domain-specific word meanings of the terms in the constituents, as shown in figure 3.6.



Semantic roles:	Agent	Action	Theme		
Syntactic constituents:	[Who]	[suggested]	[to take	[furniture]	[outside]]
Word meanings:	Person	ArgSeg:cat:propose		Topic	Topic

Figure 3.6: Lexical, syntactic and semantic analysis of the query "Who suggested to take furniture outside?"

Next, the terms and their word meanings are translated into instances of concepts in the meeting domain. For this, the NLU uses domain-specific interpretation rules. For an example, see figure 3.7.



	Input: [Term/Meaning]		**Output: [Concept.attribute=value]**	**Result for** "Who suggested...."
Rule 1	"who"/Person	→	person(n)	person1
Rule 2	Term/Arg:cat:Val	→	argseg(n).cat=Val	argseg1.cat=propose

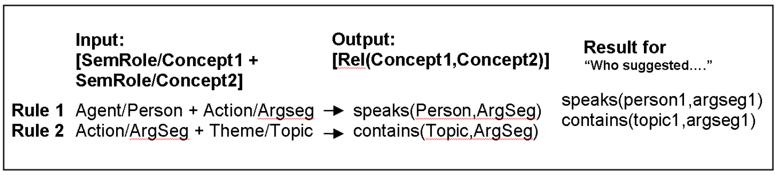
Rule 3	Term/Topic	→	topic(n).label=Topic	topic1.label=furniture
				topic2.label=outside

(n): enumerator to distinguish between multiple instances of the same concept

Figure 3.7: Concept extraction rules

The word "who" triggers the first rule, "suggested" the second rule, and "furniture" and "outside" the third rule. Notice that the instance of Person is 'anonymous' in the sense that there are no attributes that describe it. The instance of Argumentative segment is described by the attribute-value pair *category=propose*, and finally two instances of Topic are generated, each described with an attribute-value pair, *label=furniture* and *label=outside* respectively.

The third and final step of the NLU is to extract relations between the instances. This is done with a second set of domain-specific rules that use information about the semantic roles of the constituents in the linguistic query. The intuition is that linguistic semantic relations between terms, e.g. Agent / Action, correspond to given concept relations in the meeting domain, e.g. Speaker / Argumentative segment. Two example rules are given in figure 3.8.



	Input: **[SemRole/Concept1 + SemRole/Concept2]**	**Output:** **[Rel(Concept1,Concept2)]**	**Result for** **"Who suggested...."**
Rule 1	Agent/Person + Action/Argseg →	speaks(Person,ArgSeg)	speaks(person1,argseg1)
Rule 2	Action/ArgSeg + Theme/Topic →	contains(Topic,ArgSeg)	contains(topic1,argseg1)

Figure 3.8: Relation extraction rules

The first rule states that if a term in the query has the word meaning Person, and it has the role of Agent in the sentence; and if a second term has the word meaning Argumentative segment and has the role of Action in the same sentence, then the concept instances that were generated from those two terms have the relation *speaks(Person, Argumentative segment)*. Similarly, in the second rule, the semantic roles Theme and Action when assigned to two terms with meanings Topic and Argumentative segment, trigger the relation *contains(Topic, Argumentative segment)*.

The concrete meaning of the concepts and relations extracted in the above example is the following: the data to be retrieved from a meeting discussion is a proposal that was made during an episode in which the participants discussed the possibility of taking furniture outside, and the speaker of that proposal is to be identified.

The individual language processing components that are required to perform these stages of analysis are visualized in figure 3.9. The first stage, linguistic analysis, produces a representation called 'logical form' (LF) which is a compact representation of syntactic constituents, their semantic roles, and the word meanings of the terms inside those constituents. The concept extraction module extracts concept instances from the logical form. Finally, the relation extraction module adds the concept relations between the concept instances. The remainder of this sub-chapter describes the development and evaluation of these three components, including the resources used by each component.

3.5.2 Query set

To develop a domain-specific NLU module for queries we use a corpus of representative user queries. The query set has two overall purposes in the design

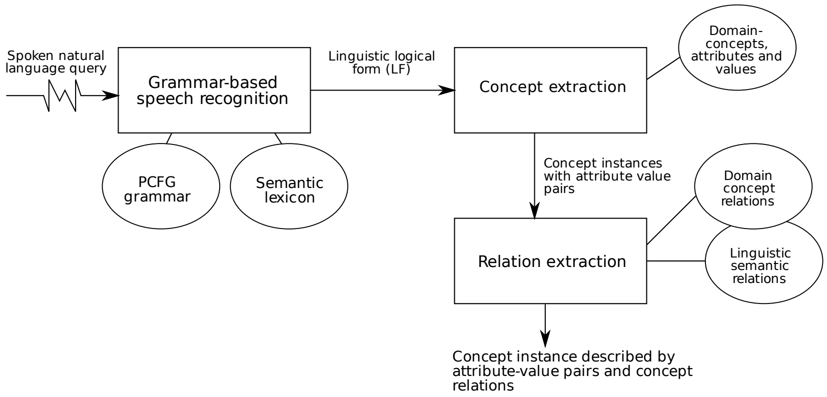

Spoken natural language query
Grammar-based speech recognition
PCFG grammar
Semantic lexicon
Linguistic logical form (LF)
Concept extraction
Domain-concepts, attributes and values
Concept instances with attribute value pairs
Domain concept relations
Relation extraction
Linguistic semantic relations
Concept instance described by attribute-value pairs and concept relations

Figure 3.9: Architecture of the natural language understanding component

of the system:

1. Provide lexical and syntactic coverage for the grammar-based speech recognizer
2. Provide syntactic and semantic patterns for the design of domain-specific concept and relation extraction rules

As representative query set we have selected a subset of 80 queries from the original 270 queries collected by (Lisowska, 2003) for user requirement analysis. More specifically, we have selected queries that refer to the topical and argumentative layers of meeting data and that can be answered by retrieving a piece of meeting discussion from a meeting recording. Since the queries in the user requirement analysis are introspective, they often contain variables rather than real terms, e.g. ”Why did X reject the proposal?” or ”What did they decide about this or that?” In order to be able to interpret such queries, they need to refer to real entities. We solved this issue by replacing variables in queries with names of persons and topics that occurred in existing meeting recordings. Notice that the goal of this work is not to evaluate the performance of an end-to-end query engine for meeting data. The queries in our set do not refer to any particular meeting recording and cannot be evaluated as such. Here we only want to extract a domain-specific interpretation of queries and evaluate its correctness with respect to the domain model.

3.5.3 Grammar-based speech recognition

The grammar-based speech recognizer, chosen for the current architecture of the natural language query engine, was implemented with the Regulus and Nuance software, in collaboration with the University of Geneva, as part of the IM2 project. Regulus is a Prolog-based toolkit for developing unification-based grammars and semantic lexica. The syntactic analysis of a sentence, parsed with a Regulus-grammar and lexicon, can be represented both as a syntactic tree and as a logical form.

Regulus provides tools for compiling a unification-based grammar and lexicon into a grammar in the Nuance Grammar Specification Language (GSL), which is then used to build the corresponding Nuance speech-recognizer. To reduce the size of the GSL grammar, which can become extremely large if a very general Regulus-grammar is used, specialization is required. Regulus provides a tool for this, implementing an Explanation Based Learning method.

The Nuance speech-recognizer analyzes the vocal input using its grammar- and lexicon-resources and at the same time parses the produced transcription. The recognition result therefore consists of both a sequence of words and a logical form (the linguistic analysis of the query). With the current choice of parameters, the recognizer returns only the 1-best result, but it is also possible to get the n-best results, with values of n ranging from 5 to 10.

3.5.4 Logical form

The logical form, provided as a result of the speech recognition, can be represented in several ways: linear forms, nested logical forms or RIACS logical forms. Without going into detail on the differences between these different representations, we choose the nested logical form, because it provides an appropriate abstraction of syntactic information for our needs. Figure 3.10 shows an example of a nested logical form. The form is a hierarchical syntactic analysis of the natural language query but not equivalent to ”standard” derivation trees produced by traditional rule-based rewriting grammars. Logical forms are more abstract, focusing on the syntactic roles of the main elements in the sentence: the subject, main verb, object etc.

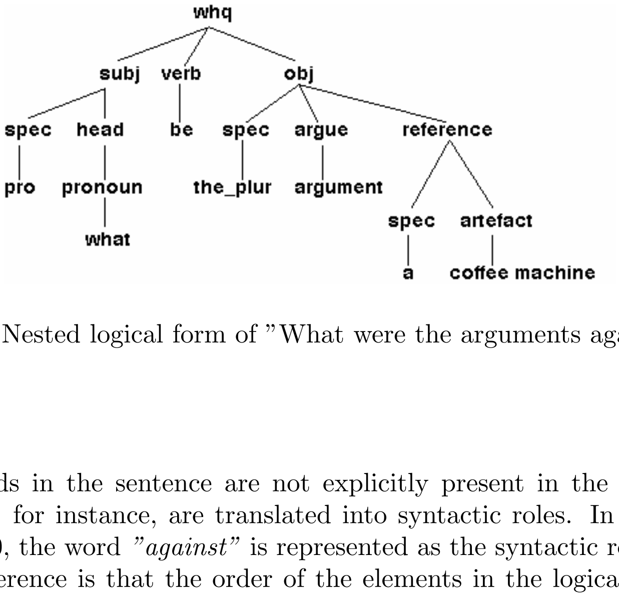


Figure 3.10: Nested logical form of ”What were the arguments against a coffee machine?”

Some words in the sentence are not explicitly present in the logical form. Prepositions, for instance, are translated into syntactic roles. In the example in figure 3.10, the word *”against”* is represented as the syntactic role *reference*. Another difference is that the order of the elements in the logical form is not dependent on the order of the words in the query. Two sentences having different word orders can be represented by the same logical form. Typical examples are active and passive constructions, as illustrated in figure 3.11.

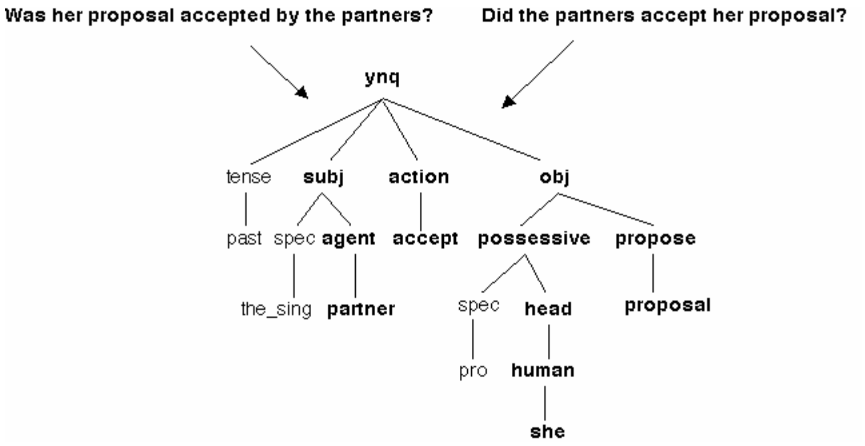


Figure 3.11: Logical form representing both active and passive construction

In addition to syntactic roles, the logical form also provides semantic information about the main words, for instance *[human,she]*. This information is expressed in terms of *semantic types*. The semantic types are specified in the lexicon, but they do not have any impact on the syntactic analysis of the query. They only provide a "semantic decoration" to the logical form. This "decoration" is however very useful for the NLU module. It helps to identify words in the query that refer to concepts in the domain model. The next section will describe how these semantic types are chosen for the lexicon in the meeting data domain.

The logical form is quite rich on syntactic and semantic information, but the NLU module does not need to access all that information to extract domain-specific concepts and relations. To simplify the access to the relevant parts, we have written a parser that transforms the nested logical form into a flat *quasi-logical form* (QLF) (see figure 3.12). This flat logical form is then processed by the concept and relation extraction modules.

In the flat QLF, each item corresponds to one node in the nested representation. In order to preserve the information about the dependencies between the nodes, a variable is added to each item, to represent its scope in the hierarchical structure. For instance, between the three items

[sentencetype,ynq,v1]
[subj,v1,v2]
[human,who,v2]

the variables *v1* and *v2* indicate that the subject is within the scope of the sentence type and that the lexicalized semantic item *[human,who]* is the subject. In the rest of the paper, logical forms will be presented in the flat QLF notation.

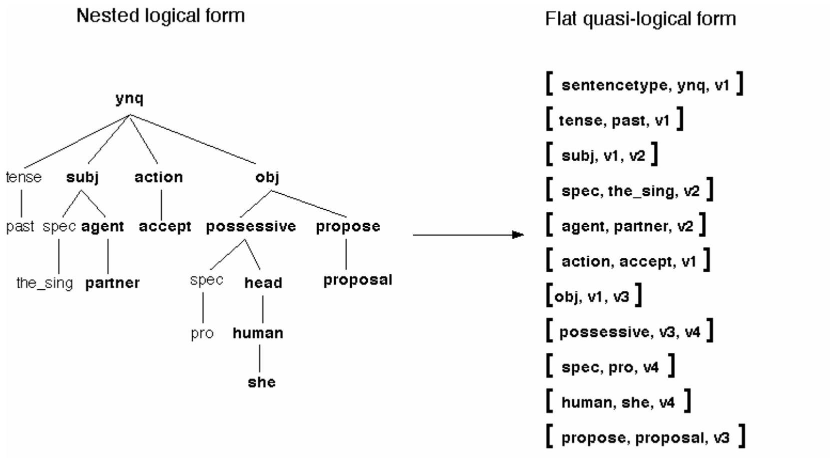


Figure 3.12: Transformation from nested logical form to flat quasi-logical form (QLF)

3.5.5 Semantic lexicon

The grammar-based speech recognizer uses two linguistic resources to recognize and analyze queries: a probabilistic context-free grammar (PCFG) and a semantic lexicon. The main purpose of the lexicon is to assign meanings to words in queries. However, being a grammar-based speech recognizer, the semantic lexicon has two purposes. The first is related to the syntactic analysis of the query. The grammar uses lexical information to restrict the possible combinations of words in the syntactic structures. For instance, only a human entity can be the subject of certain actions such as reading, presenting, talking etc. With such lexicalized semantic information the grammar can successfully analyze *"Which participants asked questions?"* while rejecting *"Which documents asked questions"*. This type of lexical information is called *sortal types* and is particularly useful during speech recognition, to prevent invalid transcriptions.

The second type of lexical information used in our system is the semantic types appearing in the logical form. These types also classify words according to their meaning, but they have no impact on the syntactic analysis and can be chosen according to the needs of the application. Either the semantic types can describe the application-specific meaning of the words, e.g. "[argseg_class, disagree]" or they can represent the general linguistic meaning, e.g. "[react, disagree]". Since the logical form is an abstract linguistic representation of the query, and the semantic lexicon is mainly used by the speech recognizer, a modular approach would be to keep linguistic and domain-specific resources separate and let the semantic types represent the linguistic meaning of the words. However, these semantic types have to provide sufficient semantic detail to enable the NLU module to map words to domain-specific concepts. The goal is to find a level of linguistic meaning that makes the mapping possible.

A large amount of work has already been done to build semantic dictionaries that organize words into conceptual hierarchies according to their meanings.

WordNet (Fellbaum, 1998) is a lexical reference system that was designed as a network, to reflect how speakers organize their mental lexicons. Synonyms are grouped together into synsets representing concepts, and concepts are linked with one another through semantic relations such as hyponymy (more specific concept) and hypernymy (more general concept).

WordNet is a well-known resource used in numerous projects and covers a large variety of synonyms for each word. We choose WordNet in an attempt to objectively classify words in the meeting domain according to general linguistic concepts. WordNet also allows us to enhance the lexicon with new synonyms that were not present in the query set. Adding synonyms to the lexicon is especially important due to the diversity of terms used for referring to meeting episodes. For example in episodes where disagreement occurred, queries that refer to that episode may contain terms and phrases like: ”have objections against”, ”reject”, ”not convinced”, ”sceptical”). In light of the fact that there are a limited, well-defined set of argumentative categories in the domain model that lexical terms can be mapped to, it is through-out possible to use resources such as WordNet to capture those variations. The classification of word meanings is done in the following way:

1. First a set of verbs and nouns are extracted from the query set that we used as basis for building the grammatical resources for the speech recognizer.
2. Second the extracted verbs are nominalized in order to obtain the corresponding noun, for example ”suggest” is nominalized to ”suggestion”. The goal is to cluster nouns and verbs as semantically equivalent, if the noun can be considered the result of the action expressed by the verb, e.g. agree-agreement, approve-approval.
3. Third, we identify synsets (linguistic concepts) in WordNet that represent the meanings of the noun-verb clusters. This creates a small ontology, a sub set of the WordNet ontology that only contains words from the meeting domain.
4. Finally, the ontology is disambiguated. If a word occurs in several synsets, only the synset relevant for the domain is kept. The other linguistic meanings of the word are removed. The deletion of synsets is done manually by comparing the different concept sub-trees in which the term occurs in WordNet, and selecting the concept tree that most closely maps to concepts in the meeting domain.

Further, with a bottom-up approach, choices are made about what nodes of a WordNet subtree to include in the lexicon. For instance, a path in a WordNet sub-tree is:

accept < **react,respond** < act,move

Two nodes in the path represent concepts that are relevant in the domain: *accept* maps to the argumentative category ”accept”, *react/respond* maps to a dual set of argumentative categories [”accept” or ”reject”]. The third concept *act/move* is too general and does not map to any concept in the meeting domain. It is therefore excluded from the lexicon.

Another example of paths in the conceptual hierarchy that are relevant to the meeting domain is two paths that contain the term "decide":

> elect<{choose, take, select}<{**decide**, make up one's mind, determine}
> **decide**< {determine,shape,mold,influence}<{cause,do,make}<{make,create}

Both meanings of the word "*decide*" are considered as relevant to the meeting domain because they can both be mapped to the argumentative category '*decide*' in the domain model. But there cannot be two meanings for the word in the semantic lexicon, so we merge the two synsets into one concept *decide*. The subnodes of *decide* in the first path (*choose, elect* etc.) are removed because they are too specific to have a meaning in the meeting domain. There is simply no annotation in the meeting records that allows for distinguishing between the two queries: "*Was there an election*?" and "*Was any decision made?*". Both queries retrieve a meeting segment that is annotated with the argumentative category '*decide*'. In order to keep the terms in our semantic lexicon, but not their too-specific linguistic meanings, we collapse all these sub-nodes into one group and let their meaning be represented by the super-concept *decide*. Similarly, we merge the two synsets {cause, do, make} and {*create, make*} to one concept *create*. The node {*determine, shape, mold, influence*} does not contain new words that are important in the domain, so this node is removed. As a result, the two paths above with their five concepts are reduced to one path with two nodes, *decide* and *create*.

> {decide, determine, choose, take, select make up one's mind} <
> {make, create, cause, do}

We have now described the linguistic processing of queries in the meeting domain: how the query is parsed, segmented into syntactic constituents, assigned with semantic roles and linguistic meanings of terms, and how these are represented in a logical form. In the next two sub-sections we describe the domain-specific interpretation of queries, and show how the syntactic and semantic information in the logical form is exploited during interpretation.

3.5.6 Concept extraction

In the first stage of domain model-driven interpretation, only the lexicalized semantic information present in the logical form is treated. The lexicalized semantic information consists of a word and its semantic type. For instance, the logical form produced for "Who suggested the coffee machine?" has the following elements (the lexicalized semantic information is marked in bold):

> [utterancetype,whq,V1]
> [subj,V1,V2]
> [spec,pro,V2]
> **[human,who,V2]**
> [tense,past,V1]
> **[propose,suggest,V1]**

[obj,V1,V3]
[spec,the_sing,V3]
[artefact,coffee_machine,V3]

The lexicalized semantic information is used for extracting instances of concepts in the meeting domain. Formally, a concept extraction rule consists of three elements, of which the second is optional:

Input: [SemType, Word]
Condition on SemType
Output: concept(Name,ID,Att,Word)

The first element of the rule specifies the input word and its semantic type, the second element specifies the possible values of the semantic type, and the last element is the concept instance produced by the rule. The four variables of the concept are: the name of the concept (Name), a unique instance identifier (ID), an attribute of this concept (Att) and finally the word in the natural language query that was mapped to this concept. For instance, the concept extraction rule for the concept *Argumentative segment* is:

Input: [SemType,W]
Condition: SemType = discuss *or* propose *or* reject *or* accept *or* ask or answer *or ...*
Output: sc(argseg,ID,class,W)

This rule applies to the logical form item [propose,suggest,V1] in the above example, and produces the concept *concept(argseg,1,class,suggest)*, where "1" is a new identifier (number) generated by the system.

Many extraction-rules only have two elements: the input and the output. These rules are common when a concept in the domain model is identified from only one semantic type. For instance:

Input: [firstname,W]
Output: sc(person,ID1,firstname,W)

Input: [familyname,W]
Output: sc(person,ID2,familyname,W)

As previously shown, some words in a user query can refer to concepts in the domain without pointing to a specific attribute, for instance "*who*", "*people*" and "*participant*", which all refer to the domain concept Person. Such words are processed by special concept instantiation rules that produce anonymous *concept instances* without attributes. Anonymous concept instances have three variables: a concept name, an instance identifier and the word that triggered the instantiation. An example of this type of rule is:

Input: [SemType,W]
Condition: SemType = human *or* person(sing) *or* person(plur) *or* agent
Output: anonymous(person,ID,W)

Anonymous concept instances do not provide any information about the specific instance of the concept, but they can play a role in the next stage of interpretation, where relations between concepts are extracted. An example of this will be shown in the next section.

There is one concept in the domain model that cannot be processed with rules of the type described above, namely the Topic concept, which represents the thematic episodes in which argumentative segments, utterances and document references occur. For instance, in the query ” *What decision was made about the sofa?*” the word ”*sofa*” should be mapped to the topic label '*sofa*'. The fact that this is a topic label is not decidable from the semantic type of the word ”sofa”. In the domain model, any word can be a topic label if the word has occurred in a meeting, either in the discussion or in a document used during the discussion.

Instead, topic labels can be recognized in queries through the context in which they appear. Typical syntactic constructions are: ”*about X*”, ”*concerning X*”, ”*discuss X*”, ”*propose X*” etc. where *X* is the topic. The rules that extract topic labels from queries therefore need to access the syntactic information in logical form rather than the lexicalized semantic information. Currently we have two general rules for extracting topic labels, one for the prepositional constructions (”*about X*”, ”*regarding X*”) etc. which represented as the syntactic relation ”reference” in the logical form, and one for the verb-object constructions ”*discuss X*” etc.:

Input : LF
Condition : [**reference**,V1,V2] [SemType,W,V2] in LF
Output : concept(topic,ID,label,W)

Input : LF
Condition : 1. [SemType1,Word1,V1] [obj,V1,V2]
[SemType2,Word2,V2] in LF
2. SemType1= **discuss** *or* **propose** *or* **utter** *or* ...
Output : sc(summary,ID,item,W2)

These two rules describe the most typical syntactic constructs appearing in queries with references to thematic episodes (topic labels in the meeting annotation), but there are cases where they do not apply. Queries, for which it is difficult to write appropriate rules, are discussed in the experimental results in section 3.5.8 .

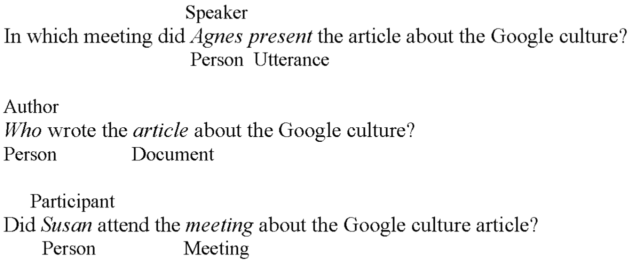
3.5.7 Relation extraction

The second stage of domain-specific interpretation is the extraction of relations between instances of domain concepts. The relations are defined by the domain model and there are two important reasons why they should be extracted from the queries.

First of all, there are concepts in the domain model that have relations with multiple concepts. Depending on which relation is referenced in the query, the instance of the concept has different *roles*. For instance, the concept *Person* has three explicit relations, and one implicit relation, with other concepts in the domain model:

Person/ Utterance: speaker
Person/ Meeting: participant
Person/Document: author
Person/Argumentative Segment: speaker

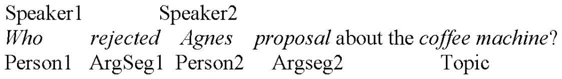
When a person is referred to in a query, the system needs to know which of these relations applies to the person, in order to be able to search for the right information in the database of meeting records. Examples of three queries that refer to three different Person-relations are:


Speaker
In which meeting did *Agnes present* the article about the Google culture?
Person Utterance

Author
Who wrote the *article* about the Google culture?
Person Document

Participant
Did *Susan* attend the *meeting* about the Google culture article?
Person Meeting

The second reason why it is important to extract relations is that a query can make reference to more than one instance of the same concept, and the instances may have different relations with the other concepts in the query. An illustrative example is the query:


Speaker1 Speaker2
Who *rejected* *Agnes* *proposal* about the *coffee machine*?
Person1 ArgSeg1 Person2 Argseg2 Topic

The query contains two references to instances of the concept Person ("*who*" and "*Agnes*") and two to the concept Argumentative segment ("*rejected*" and "*proposal*"). Extracting relations is, in this case, essential for deciding which person is the speaker of which argumentative segment.

We have now motivated why relations need to be extracted from queries in a complex domain such as the meeting domain. Concretely the extraction is done through a set of relation extraction rules, which operate in a similar way as the concept extraction rules. The difference is the relation extraction rules use syntactic rather than lexicalized semantic information to map between the linguistic and domain-specific representations of the query. Formally, a relation extracting rule has the following form:

```
Input     : concept(Concept1,ID1,Attr1,Word1),
            concept(Concept2,ID2,Attr2,Word2)
Condition : syntactic_relation(Word1,R,Word2)
Output    : relation(Concept1,ID1,Rel,ID2)
```

A relation extracting rule is triggered by particular pairs of concept instances. Each rule specifies the names of two concepts, for instance Person and Document. It also specifies if the input concept instances are anonymous or not. The output is one or several domain relations between the two input concepts. The condition, which has to hold in order for the rule to apply, is that the two words, which triggered the two concept instances, have a specific syntactic relation in the logical form. The syntactic relations are seen as possible references to the domain-specific relations. Several different syntactic relations can correspond to the same domain-specific relation.

The syntactic relations are specified in terms of syntactic roles (subject, object, reference etc.), which is a convenient level of abstraction to express how syntactic constructs in natural language correlate to domain-specific relations in the domain model. To illustrate the correlation, we give an example of a rule for the two concepts Argumentative segment and Person that accepts four different syntactic relations as correlation to the domain-specific relation *speaks*. Notice that the "speaks"-relation between Person and Argumentative segment is *implicit* in the domain model. The explicit relation is between Person and Utterance and therefore has to be expressed with two relations: 1) the instance of Person speaks the instance of Utterance, and 2) the instance of Argumentative segment contains the instance of Utterance.

```
Input     : concept(argseg,ID1,Category,Word1)
            concept(person,ID2,Attr,Word2)
Condition : syntactic_relation(Word2,possessor_of,Word1) or
            syntactic_relation(Word2,subj_relobj,Word1) or
            syntactic_relation(Word2,subj_action,Word1) or
            syntactic_relation(Word2,subj_obj,Word1)
Output    : relation(argseg,ID1,contains,ID3)
            relation(person,ID2,speaks,ID3)
```

The four syntactic relations are illustrated in figure 3.13 with four different queries, where "*Susan*" triggers an instance of the concept Person and "*decision*" an instance of Argumentative segment.

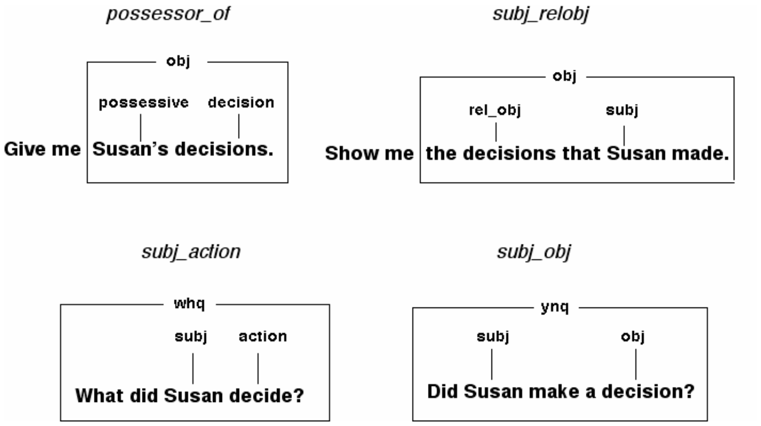


Figure 3.13: Syntactic relations that correspond to the relation speaks(Person, ArgSeg) in the meeting domain model

Notice that the above rule is not completely well-defined with respect to the syntactic relations *subject-object* and *subject-relative object*. The main verb is not taken into consideration, which means that the domain-specific relation is extracted for any verb standing between the subject and the object. For instance, "*Did Susan make a proposal?*" and "*Did Susan reject the proposal?*" are both queries where "*Susan*" is subject and "*proposal*" is object. According to the relation extracting rule, Susan is the speaker of the proposal in both queries, which is not the case. Another example where the verb is important is "*Did Susan write an article?*" and "*Did Susan present an article?*" where the verb decides if Susan is the *speaker* of an utterance or the *author* of a document. To incorporate relevant information about the verb in the subject-object relations, we add a variable to these relations, specifying the semantic type of the verb. For instance, in the previous rule, we modify the subj-obj and subj-relobj conditions to:

VerbSem = utter *or* inform *or* create *or* activity *or* provide
syntactic_relation(Word1,subj_obj,Word2,VerbSem)
syntactic_relation(Word1,subj_relobj,Word2,VerbSem)

With this restriction imposed on the semantic type of the verb, the rule applies for subject-object relations only when the verb is: make, have, present, express, provide, give etc. but not when it is write, accept or reject.

A syntactic relation is validated by identifying a specific syntactic pattern between two words in the logical form. The concrete pattern depends on the type of syntactic analysis available, e.g. logical form or compact tree forest. In a logical form, the syntactic roles of the words are provided in an explicit way, which makes it very easy to specify the precise pattern that corresponds to a syntactic relation. For instance, the possessor_of relation can be expressed with the following pattern:

```
syntactic_relation(Word1,possessor_of,Word2)        : [possessive,V1,V2])
                                                      [SemType1,Word1,V2]
                                                      [SemType2,Word2,V1]
```

This pattern occurs in " *What was Susan's decision?*" where "*Susan*" is the possessor of "*decision*". The logical form for this query is illustrated below and the parts corresponding to the syntactic pattern of the possessor_of relation are marked in bold:

[utterancetype,whq,v1]
[subj,v1,v2]
[spec,pro,v2]
[pronoun,what,v2]
[tense,past,v1]
[exist,be,v1]
[obj,v1,v3]
[possessive,v3,v4]
[spec,pro,v4]
[firstname,susan,v4]
[decide,decision,v3]

For subject-object relations, where the semantic type of the verb needs to be specified, the pattern is expressed as:

```
syntactic_relation(Word1,subj_obj,Word2,VerbType) : [subj,V1,V2])
                                                    [SemType,Word1,V2]
                                                    [obj,V1,V3]
                                                    [SemType,Word2,V3]
.                                                   [VerbType,Verb,V1]
```

The advantage of encoding the syntactic patterns into separate rules (or "functions") instead of specifying them directly in the relation extraction rules is that it makes the processing of the queries very modular. If the syntactic representation of the query is changed, for instance to standard syntactic derivation trees, the relation extraction rules do not have to be modified. Only the descriptions of the syntactic patterns need to be updated. The modularity is relevant with respect to the choice of speech recognition technology. Grammar-based speech recognition works well on small scale, but if the system is scaled to a larger application, a more robust natural language processing architecture may be needed that produces a forest of (partial) derivation trees.

3.5.8 Functional validation

A prototype of the NLU module was implemented in Prolog according to the architecture in 3.5.1. The concept- and relation-extraction rules were designed based on the query set in 3.5.2.

The implementation resulted in 25 concept extraction rules and 30 relation extraction rules. 11 syntactic relations were used by the relation extraction rules. Some of them, not mentioned in the previous examples, are *subject-action*, *action-object*, *reference*, *of*, *to* and *from_location*. The reason why the syntactic relations are relatively few in comparison to the concept relations in the domain model is that the same syntactic relation can apply to multiple concept relations.

The lexicon used by the speech recognizer was implemented in collaboration with the University of Geneva and was still under development when the evaluation of NLU engine was performed, which means that the semantic types of lexical terms described in section 3.5.5 were not yet fully implemented in the lexicon and were not available at the time of testing. To be able to do experiments, another ad-hoc semantic lexicon was created in Prolog, which simply listed all the words in the lexicon with their corresponding word meaning, some of which were domain-specific: *first name, family name, meeting, document, person, institute, argumentative category, create, communicate, provide,* etc.

Also the grammar was not yet completed, which means that some of the queries in the evaluation received no logical form, and therefore could not be processed by the subsequent concept extraction and relation extraction modules. Variations of such queries, compatible with the current version of the grammar, were tested instead.

The goal of the evaluation was to verify that the concept and relation extraction rules were well-formed and that the syntactic relations were appropriate for constraining the extraction of domain-specific relations from queries. The 80 queries in the corpus were parsed with the available version of the grammar, and the 1-best linguistic analysis was then processed by the two domain-specific interpretation modules. Speech recognition errors were not considered in this evaluation. Only written queries were tested.

To measure precision and recall of the NLU engine, concepts and relations were extracted manually from all the queries in the corpus to serve as reference. The automatically extracted concepts and relations were then divided into four groups according to the following levels of precision and recall:

1. Full extraction: All concepts and relations were extracted correctly
2. Partial extraction: Some of the concepts or relations in the reference were not extracted automatically
3. Incorrect extraction: Some of the concepts or relations that were extracted automatically were not found in the reference
4. No extraction: The NLU module did not extract any concepts from the query

For the queries that have no result, a distinction was made between cases where the syntactic analysis failed to produce a logical form, and where the NLU module failed to extract concepts.

The results are presented in table 3.6. Among the 80 queries that were tested, there was no case of an incorrect extraction or cases where the NLU module failed to extract something from a syntactic analysis that was provided. On the other hand, almost half (32) of the queries in the test corpus had no syntactic analysis. This was mainly due to the implementation status of the grammar. In many cases, the parse failure could be overcome by changing a single word in the query. To be able to test the domain-specific interpretation on these queries, we made minor modification to queries to obtain a syntactic analysis. Typical modifications were changing a preposition for another (e.g. "*on*" to "*of*"), or a proper noun for a pronoun (e.g. "*Susan's opinion*" to "*her opinion*"). These changes were considered to have no important effect on the concept and relation extraction, as the logical form would be almost identical and the number of extractions would remain the same. A third type of change was to remove a prepositional phrase, e.g. by changing "*Who suggested the solution to solve the problem with the white board?*" to "*Who suggested the solution to solve the problem?*" This type of change was applied in five cases, and it simplified the query by reducing the number of concepts that could be extracted.

	Full extraction	Partial extraction	No analysis
Original 80 query set	36%	24%	40%
Modified 80 query set	59%	28%	14%

Table 3.6: Results of domain-specific concept and relation extraction

From the 32 queries that had no syntactic analysis, altogether 21 were modified so that a logical form was obtained and the query processed by all NLU modules. The remaining 11 queries were left unchanged because there was no simple modification that could be made to enable syntactic analysis.

When observing the queries that received full extraction of concepts and relation from the NLU engine, it is important to note that there are quite substantial differences in the level of complexity of these queries. The most simple ones (28%) are those that produce a single semantic constraint, e.g.

What decisions were made?	concept(argseg,1,category,decision)

Queries of medium complexity (38%) map to two concepts and one relation, e.g.

Who participated to the meeting?	concept(person,1,who)
	concept(meeting,2,meeting)
	relation(meeting,2,participant,1)

The most complex queries involve at least three concepts and two relations. 34% of the successfully processed queries belonged to this type. For those queries that received a partial extraction, the distribution was 47% medium and 53% complex. The numbers indicate that it is not the *complexity* of the query that represents the difficulty in extraction. To understand what other factors are involved, we looked more closely at the queries that received a partial analysis. Several problems were identified.

Returning to the implementation of the relation extraction rules, a strong assumption was made about the correlation between syntactic relations in the query and domain-specific concept relations in the domain model. The assumption is that a relation is instantiated by two words (representing two concepts) that are connected with a syntactic pattern, e.g. subject-action. In some queries in the test corpus, however, single words can refer explicitly to a concept but at the same time implicitly to a relation. For instance, the word *participant* refers explicitly to the concept Person and implicitly the relation *participates* between Person and Meeting. A query such as " *Who were the participants?*" should give the following semantic constraints:

concept(person,1,who)
relation(meeting,2,participant,1)

Instead the concept and relation extraction rules produce:

concept(person,1,who)
concept(person,2,participant)

Since there is no word in the query referring explicitly to the concept Meeting, the participant-relation currently cannot be extracted. Additional, non-binary, rules need to be designed to account for this phenomenon.

A second problem is the extraction of instances of Topic and the different relations they have with other concepts in the query. In the current implementation, there are two rules that identify instances of Topic based on simple syntactic patterns: verb-object (e.g."*discuss X*") and reference (e.g. "*about X*"). These rules are quite sufficient if the Topic instance is represented by a single word or a short noun phrase. But the rules are not able to handle cases such as "*Was someone opposed to having a computer?*" where the syntactic pattern covers "*opposed to X*", and *X* is a longer phrase, for example a verb phrase, that contains the terms that refers to the Topic instance, in this case the term "*computer*". Some more sophisticated rules are required for this type of queries.

Even when a Topic instance is successfully identified, it is not always the case that the relation between the Topic instance and another concept in the query is extracted properly. Relations are extracted with the same syntactic patterns as Topic instances, i.e. "Concept1 *references* Concept2" or "Concept1 is in *verb-object* relation with Concepts2". When there is ambiguity in the linguistic analysis, one of the analyses is chosen arbitrarily as the input for the subsequent NLU modules. Sometimes the "wrong" analysis is chosen, which means that the syntactic relation checked by the NLU module does not exist in the logical form, although it exists in one of the other non-chosen analyses. For instance, the query "*What was the outcome of the discussion about the sofa?*"

is ambiguous with respect to the scope of the prepositional phrase "*about the sofa*". Two segmentations are possible:

> [What was the outcome [of the discussion [about the sofa]]]
> [What was the outcome [of the discussion] [about the sofa]]

To illustrate the problem of random selection of linguistic analysis, we describe the steps of domain-specific interpretation. In the first step, the concept extraction module generates two concepts from the logical form, regardless of which of the two above segmentations is being processed: an Argumentative segment ("discussion") and a Topic ("sofa"). In the next step, a relation extraction rule tries to extract the relation *contains(Argumentative segment, Topic)* by checking if the appropriate syntactic relation exists between "*discussion*" and "*sofa*", in this case "*discussion references sofa*". In the linguistic analysis of the first segmentation this relation exists, in the second segmentation it does not. Consequently, the correctness of the syntactic analysis determines if a domain-specific concept relation is extracted or not. Therefore, some care has to be taken in selecting the "right" syntactic analysis. Alternatively all analyses (if there are not too many) may be processed in parallel and potential conflicts between the interpretations then resolved.

3.6 Conclusion of the chapter

In this chapter we addressed two aspects of natural language querying of meeting discussions - the need for argumentative structuring of meeting discussions to enable efficient querying of the data, and the techniques required for natural language understanding of questions to correctly map terms in queries to concepts and relations in the domain model of meeting data. By analyzing query collections, annotating meeting transcripts, and implementing a model-driven natural language understanding module, we made the following important findings:

- The majority of questions about meeting discussions rely on argumentative structuring of the discussion in order to be answered

- Annotating a meeting discussion with argumentative categories is hard. Most argumentative contributions in a discussion can be interpreted as belonging to more than one category. The inter-annotator agreement will always be low, if the task is to assign a unique category to each argumentative segment of the discussion.

- From a querying point of view, argumentative contributions in discussions *should* be assigned with several categories, if all of them lead to relevant answers to questions.

- From a natural language point of view, queries are often complex and refer to multiple concepts and relations in the meeting domain model. A nave word-spotting algorithm for interpreting questions is not sufficient.

- A two-step approach to natural language understanding, with linguistic analysis of the question, followed by domain-specific rule-based interpretation, works well in this domain. In particular, syntactic role assignment to terms in the query is a powerful and simple linguistic tool extracting concepts and relations in the meeting domain. A relatively small set of rules is sufficient to interpret a wide range of questions. The assumption that all concept relations have a corresponding syntactic relation in queries is a close approximation but not fully supported. In natural language meanings can often be implicit. Special rules are required for capturing these.

It is important to note that the work on argumentative annotation of discussions and understanding of questions was based on a query collection that was elicited through a questionnaire. It is commonly acknowledged that such survey studies provide important requirements for the development of system prototypes but are not fully representative of real questions that users would ask if they were interacting with an actual computer. Our next objective is therefore to evaluate users' interactions with a real system prototype, and based on the interactions with that system determine what efforts need to be made for annotating discussions with argumentative categories and for natural language understanding of questions. The following two chapters will address these issues.

4 Multimodal search in meeting discussions

This chapter describes the user evaluation of a multimodal graphical user interface for meeting data retrieval. The objective of the evaluation is to gain insights about the natural language interaction with such a system, in particular whether users exploit the argumentative annotations made on the discussions when expressing queries, what the natural language understanding requirements for interpreting these queries are, and how natural language enhances the interaction with a graphical user interface in this specific domain. The evaluation was performed as a Wizard of Oz experiment, and users were given access to different combinations of the interaction modalities speech, mouse and keyboard. Results show that argumentative annotations made on discussions are exploited less than expected. Also, queries tend to be short and lack the type of structure that allows for rule-based interpretation of syntactic roles. Instead, queries are sequential. When comparing linguistic, non-linguistic and combined versions of the system, natural language enhances the interaction with the graphical user interface, by providing both efficiency and pleasure in using the system. We conclude that natural language is highly appropriate for querying meeting data, but that the natural language understanding component of the system needs to deal with both deep linguistic analysis of complex full-sentence questions, and contextual understanding of simple, sequential queries. Contextual information is, for example, the history of queries and manipulations of the graphical user interface. We also conclude that users need more concrete support and examples of the topical and argumentative structuring of meeting discussions in order to be able to take advantage of these annotations and ask more targeted questions in this relatively novel domain of information retrieval.

4.1 Introduction

An important part of the research on meeting data storage and retrieval is the development and evaluation of user interfaces for accessing this data. In chapter 3 we considered a natural language query engine that allows users to ask questions about specific episodes in the discussion. By matching the questions to concepts in a formal domain model of meeting discussions the corresponding episodes can be retrieved. Our work showed that such a system is technically possible to develop, but there was no evaluation of an end-to-end prototype involving real users. Moreover, this type of user interface is only one of many examples of how meeting data can be accessed. In the literature, at least three types of user interfaces for accessing meeting data occur, which are generally referred to as 'meeting browsers' (Bouamrane and Luz, 2007). These systems focus on the different data that are generated from recorded meetings. The first type is the audio-centric browser (Wellner *et al.*, 2004), in which the user can play the audio recording of the meeting in different speeds to get through the meeting faster. The second type is the document-centric browser that for example aligns meeting episodes with slides from (powerpoint) presentations that were referred to during the meeting (Lalanne *et al.*, 2004). The third type is the transcript-based meeting browser (Popescu-Belis and Gorgescul, 2006) where the user searches in the transcript by providing different search criteria, typically about the topics, speakers and arguments in the meeting. It is not evident which type of browser allows for the most efficient access to meeting data, or which interface design most appeals to users, because these browsers specialize on different types of information need. For example if the question is about the moods of the participants the most efficient way to access that information is by watching the recording. If the question, on the other hand, is about an episode where the participants discussed something visual that was projected with a slide, such as the dimensions of a room and how to place furniture in it, then the document-centric browser is likely to be the most suitable for answering the question. Studies that compare meeting browsers, such as the BET evaluation test (Wellner *et al.*, 2005), are meaningful as long as they compare systems specializing on the same type of data. There is to our knowledge no meeting browser that claims to provide efficient access to all the different data associated with meetings.

This chapter addresses the third type of meeting browser, the transcript-based. We evaluate a multimodal graphical user interface, Archivus, which allows users to interact both with natural language (speech or keyboard) and with tactile manipulation. The interface is flexibly multimodal, which means that it offers users the possibility to switch between modalities and for any action select the modality that is most natural to them. The evaluation has two distinct goals.

The first goal is to collect 'real' user queries about meeting discussions, as opposed to queries collected through user requirement questionnaires. This data is important for two reasons:

1. The data reflects which meeting annotations are indeed exploited in the natural language queries and provide validation for the need of argumentative structuring of meeting discussions, as described in 3.4, to answer questions
2. The data represent the true linguistic phenomena that the natural language understanding module needs to deal with and provide validation for the need of a rule-based interpretation of queries that relies on lexical semantics and semantic role extraction, as described in 3.5.

The second goal of the evaluation is to determine the appropriateness of natural language for the task, as opposed to a standard graphical user interface. First, it allows us to validate that natural language is indeed the preferred modality for searching in transcribed meeting discussions. Second, it allows us to identify situations in which users choose not to use natural language but perform tactile manipulation instead. For the selection of search criteria, for example, we study in which cases users choose to select criteria from lists, columns or clickable maps, as opposed to cases where they choose to simply make a query in natural language. This gives valuable guidance for how to efficiently integrate natural language and tactile manipulation in a multimodal graphical user interface.

The contents of this chapter are structured as follows: section **4.2** describes the multimodal transcript-based **meeting browser** Archivus that was developed at the Artificial Intelligence Laboratory at EPFL in collaboration with the university of Geneva; the library metaphor that was implemented in the design of the user interface; the interaction modalities and how to switch between them; and the interpretation of multimodal input into attribute-value pairs. Section **4.3** explains the **evaluation framework** based on the Wizard of Oz method for natural language interfaces, and the extensions made to the technical setup, software and wizard tasks, to enable evaluation of multimodal graphical user interfaces. Section **4.4** details the **research questions and hypotheses** associated with the two research goals listed above. Section **4.5** describes the **experimental conditions** used in the evaluation based on access to different interaction modalities, the evaluation procedure including all the documents that users received and the evaluation phases that they went through, and the task which consisted of answering a set of questions. In section **4.6** we present **experimental results** showing that users favour topic and keyword criteria over novel types of criteria when expressing queries, that the natural language requests are often short and lack linguistic structure, but that users perform many navigational actions in natural language that were foreseen to be performed with tactile manipulation, and that the performance on the task is higher when natural language is the dominating modality. Section **4.7** **concludes** that natural language is appropriate for searching in meeting discussions, but that users do not take sufficiently advantage of the annotations made on this data, and that some means should be found to incent users to expand their natural language search on this data.

4.2 The Archivus system

4.2.1 Related research goals

Archivus is a multimodal dialogue system for meeting data retrieval and browsing that was developed as part of the Swiss federal Interactive Multimodal Information Management (IM2) project. Several research goals are associated with the design and development of this system.

On a functional level, Archivus operates as a natural language dialogue system, implementing a mixed-initiative dialogue model (Melichar *et al.*, 2006). By integrating a graphical user interface into the dialogue system, the interaction becomes multimodal. The design implications of such an extension to natural language dialogue systems are reported in the PhD thesis of Melichar (2008).

On an interactional level, Archivus is a language-enabled graphical user interface that allows users to choose, at any given time, the modality that they find most natural for performing a task. Factors that influence modality choices and switches from one to another are reported in the PhD thesis of Lisowska (2007).

On an informational level, Archivus is a search engine to meeting data that allows users to express queries either in natural language or by selecting search criteria from menus. The main objective of this thesis is to investigate the role of natural language in this search domain, how users choose to search when speaking, how it enhances the search, and what natural language understanding requirements it poses on the development of natural language processing components of the system (Ailomaa *et al.*, 2006).

4.2.2 The meeting data

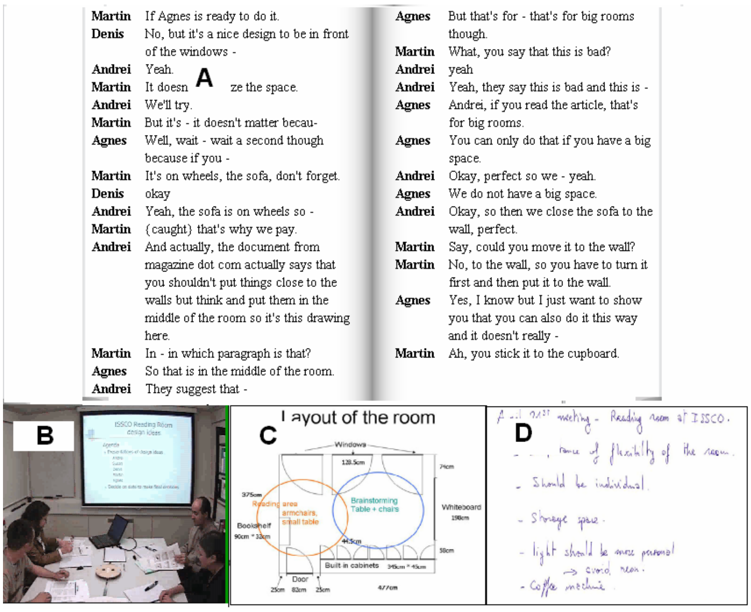
Archivus is a transcript-based meeting browser, which means that it is first and foremost intended for searching in meeting transcripts. The smallest unit of search is an utterance. Each utterance is associated with a speaker, keywords, topic, and dialogue act. If an utterance is argumentative, it is also associated with an argumentative category, or if an utterance refers to a document, then it is associated with a document reference.

However, a meeting transcript alone is not sufficient to capture all the information that occurs in a discussion. Phenomena such as moods, irony, direction of gaze, level of attention, gestures and overlapping speech are missed, and such phenomena may change the meaning of what was said. Therefore, Archivus also provides the original video recording of the meetings and aligns it with the transcript, so that when a user finds an episode of interest in the transcript, they can play the video starting at that episode.

Archivus also contains the original documents that were created or presented in a meeting, such as powerpoint presentations and hand-written notes. It does not provide the possibility to search within documents, for example with keywords, but it aligns the documents with the meeting discussion, so that if a user is reading an episode of the transcript that relates to a document, there is a link to that document in the transcript, which permits the user to browse through it. For examples of accessible meeting data in Archivus, see figure 4.1.

4.2.3 The graphical user interface

The design of the graphical user interface in Archivus is based on a library metaphor. Studies have shown that metaphors reduce the cognitive load of learning the system functionalities and offer more intuitive interaction (Cheon, 2008). For example, the metaphor of folders and subfolders for storing and browsing electronic documents is easier to learn for inexperienced computer users than a command line interface. In the case of natural language enabled user interfaces, it is particularly interesting to implement metaphors as they may help users to understand what they can say to the system. Searching in


A
B
ISSCO Reading Room design ideas
Agenda
- Presentations of design ideas
- Decide on date to make final decisions
C
Layout of the room
Windows
128.5cm
74cm
375cm
Reading area armchairs, small table
Brainstorming Table + chairs
Whiteboard
198cm
Bookshelf
90cm * 32cm
44.5cm
58cm
Built in cabinets
345cm * 45cm
Door
25cm
82cm
25cm
477cm
D
April 21st meeting - Reading room at ISSCO.
- ..., [illegible] of flexibility of the room.
- Should be individual.
- Storage space.
- light should be more personal
→ avoid neon.
- Coffe machine.

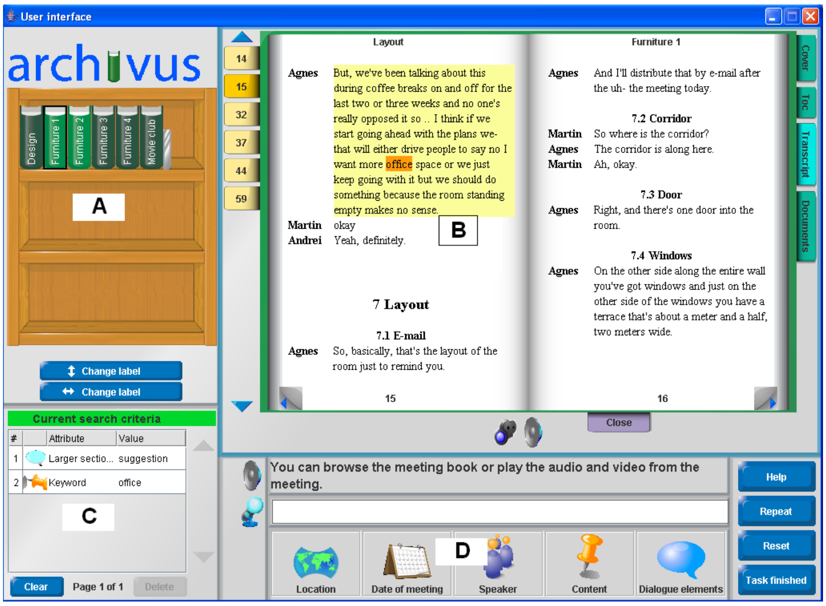
Martin	If Agnes is ready to do it.
Denis	No, but it's a nice design to be in front of the windows -
Andrei	Yeah.
Martin	It doesn ze the space.
Andrei	We'll try.
Martin	But it's - it doesn't matter becau-
Agnes	Well, wait - wait a second though because if you -
Martin	It's on wheels, the sofa, don't forget.
Denis	okay
Andrei	Yeah, the sofa is on wheels so -
Martin	{caught} that's why we pay.
Andrei	And actually, the document from magazine dot com actually says that you shouldn't put things close to the walls but think and put them in the middle of the room so it's this drawing here.
Martin	In - in which paragraph is that?
Agnes	So that is in the middle of the room.
Andrei	They suggest that -
Agnes	But that's for - that's for big rooms though.
Martin	What, you say that this is bad?
Andrei	yeah
Andrei	Yeah, they say this is bad and this is -
Agnes	Andrei, if you read the article, that's for big rooms.
Agnes	You can only do that if you have a big space.
Andrei	Okay, perfect so we - yeah.
Agnes	We do not have a big space.
Andrei	Okay, so then we close the sofa to the wall, perfect.
Martin	Say, could you move it to the wall?
Martin	No, to the wall, so you have to turn it first and then put it to the wall.
Agnes	Yes, I know but I just want to show you that you can also do it this way and it doesn't really -
Martin	Ah, you stick it to the cupboard.

Figure 4.1: Types of meeting data that can be accessed with the Archivus system: (A) Meeting transcript (B) Meeting recording (C) Referenced documents (D) Hand-written notes

meeting discussions is a relatively novel task, and there are no real-world archives of meeting data that can be translated into a "meeting data metaphor" in the user interface. Libraries, however, archive information in general, and there is a close enough resemblance between the two to find mappings between concepts in libraries and concepts in virtual meeting archives.

In Archivus, meetings are represented as books that are stored in a bookcase according to some logical order (see figure 4.2). By default they are ordered alphabetically by meeting title which is shown on the spine of the book. But being a virtual library, the system has the possibility to reorder books according to any parameter that helps users to find the books they want, such as the date or place of the meeting, or the participants. This feature is useful when there are many meetings in the database and only some of them can be displayed on the interface. Another feature of the bookcase is that is shows which books are relevant to the search by highlighting them in a lighter shade and making them bounce for a short while to attract attention. This is useful when more than one meeting matches the search criteria. The system can only open one book at a time, and when the search criteria lead to a unique meeting book, the system can open it automatically, whereas in the case of several matches, the user needs to choose which one to open.

Inside the book, the meeting data is organized into four sections that can be accessed with green content tabs (see figure 4.3). The cover page contains the date and place of the meeting, and the names of the participants. The table



Layout

Agnes But, we've been talking about this during coffee breaks on and off for the last two or three weeks and no one's really opposed it so .. I think if we start going ahead with the plans we- that will either drive people to say no I want more office space or we just keep going with it but we should do something because the room standing empty makes no sense.
Martin okay
Andrei Yeah, definitely.

7 Layout

7.1 E-mail

Agnes So, basically, that's the layout of the room just to remind you.

15

Furniture 1

Agnes And I'll distribute that by e-mail after the uh- the meeting today.

7.2 Corridor

Martin So where is the corridor?
Agnes The corridor is along here.
Martin Ah, okay.

7.3 Door

Agnes Right, and there's one door into the room.

7.4 Windows

Agnes On the other side along the entire wall you've got windows and just on the other side of the windows you have a terrace that's about a meter and a half, two meters wide.

16

#	Attribute	Value
1	Larger sectio...	suggestion
2	Keyword	office

You can browse the meeting book or play the audio and video from the meeting.

Figure 4.2: The graphical components of the Archivus system: (A) Bookcase (B) Open book (C) Current search criteria list (D) Search criteria buttons

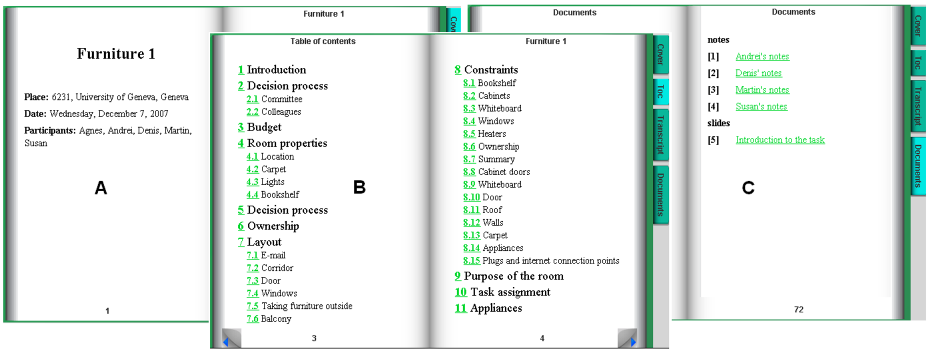
of contents gives an overview of the topics that were discussed by showing the complete hierarchical topical segmentation of the meeting discussion. The table of contents is interactive and allows users to access a topical episode of the discussion directly by selecting it instead of browsing through the transcript.

The transcript is accessed with the third content tab. If no search criteria have been specified, the transcript is a plain text that can be browsed page by page. When search criteria are present, the relevant pages of the meeting are marked with yellow hit tabs and can be browsed sequentially, skipping the pages that are not relevant to the search. On the hit pages, the relevant episodes of discussion are highlighted in yellow. If the search criteria contain keywords, these are highlighted in orange in the transcript.

The appendix contains all the documents that were presented or created during the meeting. They are organized by type: first the meeting agenda, then powerpoint presentations, papers and articles, and finally drawings on the whiteboard and hand-written notes made by the participants. Documents that were presented in the meeting can also be accessed from within the transcript, but documents that were dynamically created during the meeting, such as the handwritten notes, can only be accessed through the appendix.

4.2.4 The search criteria

To search in the meeting transcripts, a fixed set of search criteria are available in the system. These criteria have been extracted from the meeting data



Furniture 1

Place: 6231, University of Geneva, Geneva

Date: Wednesday, December 7, 2007

Participants: Agnes, Andrei, Denis, Martin, Susan

1

Furniture 1

Table of contents

1 Introduction
2 Decision process
2.1 Committee
2.2 Colleagues
3 Budget
4 Room properties
4.1 Location
4.2 Carpet
4.3 Lights
4.4 Bookshelf
5 Decision process
6 Ownership
7 Layout
7.1 E-mail
7.2 Corridor
7.3 Door
7.4 Windows
7.5 Taking furniture outside
7.6 Balcony

3

Furniture 1

8 Constraints
8.1 Bookshelf
8.2 Cabinets
8.3 Whiteboard
8.4 Windows
8.5 Heaters
8.6 Ownership
8.7 Summary
8.8 Cabinet doors
8.9 Whiteboard
8.10 Door
8.11 Roof
8.12 Walls
8.13 Carpet
8.14 Appliances
8.15 Plugs and internet connection points
9 Purpose of the room
10 Task assignment
11 Appliances

4

Documents

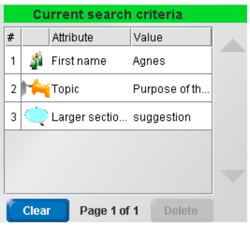
Documents

notes
[1] Andrei's notes
[2] Denis' notes
[3] Martin's notes
[4] Susan's notes
slides
[5] Introduction to the task

72

Figure 4.3: Sections of the book accessible with content tabs: (A) Cover page (B) Table of contents (C) Appendix

and annotations in the system's database. For example, the possible values for keywords are extracted from the words that were uttered in the meeting discussions. Users can select among the available search criteria by accessing the buttons at the bottom of the screen, or by speaking or writing a natural language query. In the case of natural language, the system interprets the query and generates the most probable set of criteria based on the available values in the database. In some cases queries do not match any values, and the system tells this to the user. When queries do generate criteria, these are displayed in the Current search criteria list at the far left of the screen. If several queries are made consecutively, the generated criteria represent the conjunction of the queries. For example, the two requests "Show me the discussion about the purpose of the room" and "Show me Agnes's suggestions" are equal to "Show me Agnes's suggestions about the purpose of the room". The generated criteria are shown in figure 4.4.



#		Attribute	Value
1		First name	Agnes
2		Topic	Purpose of th...
3		Larger sectio...	suggestion

Figure 4.4: Current search criteria list, showing the interpretation of "Show me Agnes's suggestions about the purpose of the room"

There are five groups of search criteria in the system. The first two, date and location criteria, allow for searching for meetings based on where and when they occurred. The third group is speaker criteria, which allow for finding meetings and episodes in which a specific person talked. The fourth is the content criteria which are divided into topics and keywords. The topic criteria are used when searching for larger episodes of a meeting where the topic was discussed. Keywords are used for finding specific utterances in the meeting where the word was explicitly expressed. To the content criteria belong also referenced documents, which are described by their title and document type, e.g. slides or pdf. The last group of criteria is the dialogue elements. This group represents the discourse annotations made on the meeting discussions. It contains both dialogue acts and argumentative categories. For non-expert users, these criteria are the most novel ones for searching in textual data.

When search criteria are accessed with the search criteria buttons at the bottom of the screen, the attributes and values are accessed hierarchically. For example, when a user selects the Speaker criteria button, the top-level Speaker criteria menu is displayed, which allows the user to specify all attribute-value pairs for a given speaker (firstname, lastname, etc.) by selecting a row in the menu (see figure 4.5). If the user prefers to select only one attribute-value pair, e.g. the speaker's first name, they can access the blue attribute-buttons (circled in red) which then open a new menu with the values for that specific attribute. The Date criteria are organized in the same manner.

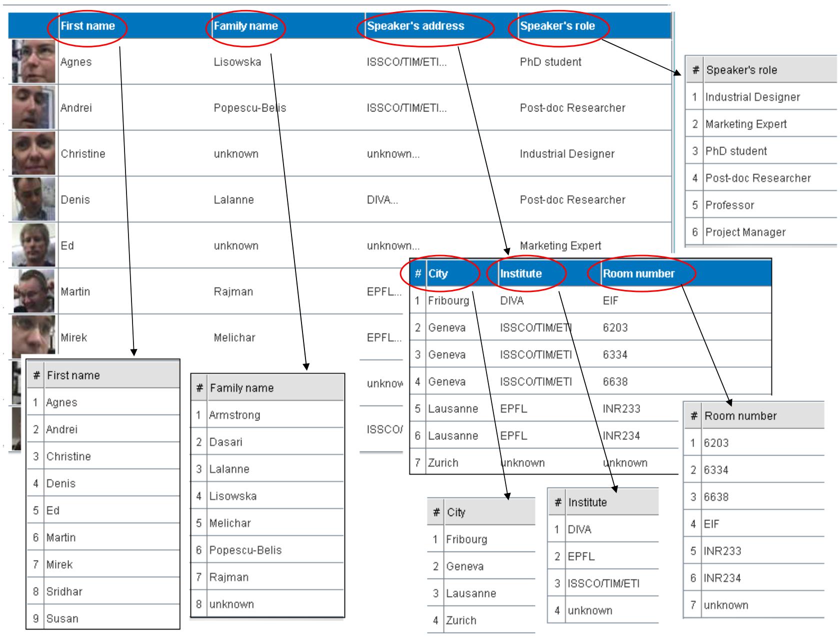
The Content and Dialogue elements criteria are also organized hierarchically, but with a difference principle (see figure 4.6). For example, when the user accesses the Content criteria button, the next step is to select one of the sub categories Topic, Keyword or Document. For each sub category, examples of values are shown to give an overview of what type of search criteria they represent. When a sub category has been selected, an alphabetical list of values is displayed.

Location criteria are the only ones which are not represented hierarchically. Instead of menus, the locations are visualized as cities on a map (see figure 4.7). The design works well when there is only one meeting place for every city stored in the database, as is the case of the current Archivus database. If multiple meeting locations are in the same city, also these criteria have to be accessible hierarchically, e.g. by first selecting the city and then the institution.

4.2.5 The multimodal interaction paradigm

Archivus is flexibly multimodal, which means that any interaction with the system can be done with any of the available modalities speech, keyboard or mouse. The motivation for such a design is that it allows us to evaluate the system without applying any prior assumptions about what modality is most optimal for performing a given task. If the user is free to choose, it is assumed that they will choose the modality that they find most natural for the task.

In practice, the flexible multimodality enforces two rules: first that for any natural language interaction there must be a series of tactile interaction that is equivalent in meaning. For example if a user makes a request that translates into three search criteria, those criteria must also be selectable from menus. Second, all tactile interactions must also be possible to make as natural language commands. This means that not all natural language interactions have to be



	First name	Family name	Speaker's address	Speaker's role
	Agnes	Lisowska	ISSCO/TIM/ETI...	PhD student
	Andrei	Popescu-Belis	ISSCO/TIM/ETI...	Post-doc Researcher
	Christine	unknown	unknown...	Industrial Designer
	Denis	Lalanne	DIVA...	Post-doc Researcher
	Ed	unknown	unknown...	Marketing Expert
	Martin	Rajman	EPFL...	
	Mirek	Melichar	EPFL...	
			unknow	
			ISSCO/	

#	Speaker's role
1	Industrial Designer
2	Marketing Expert
3	PhD student
4	Post-doc Researcher
5	Professor
6	Project Manager

#	City	Institute	Room number
1	Fribourg	DIVA	EIF
2	Geneva	ISSCO/TIM/ETI	6203
3	Geneva	ISSCO/TIM/ETI	6334
4	Geneva	ISSCO/TIM/ETI	6638
5	Lausanne	EPFL	INR233
6	Lausanne	EPFL	INR234
7	Zurich	unknown	unknown

#	First name
1	Agnes
2	Andrei
3	Christine
4	Denis
5	Ed
6	Martin
7	Mirek
8	Sridhar
9	Susan

#	Family name
1	Armstrong
2	Dasari
3	Lalanne
4	Lisowska
5	Melichar
6	Popescu-Belis
7	Rajman
8	unknown

#	City
1	Fribourg
2	Geneva
3	Lausanne
4	Zurich

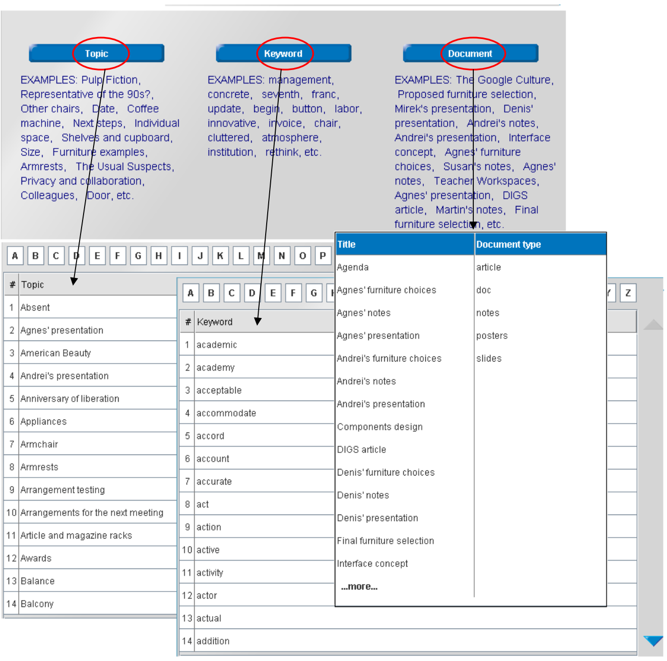
#	Institute
1	DIVA
2	EPFL
3	ISSCO/TIM/ETI
4	unknown

#	Room number
1	6203
2	6334
3	6638
4	EIF
5	INR233
6	INR234
7	unknown

Figure 4.5: Hierarchical access to speaker criteria menus

questions, as in the case of the query engine in chapter 3. Natural language can also be used for manipulating the graphical user interface, for example by saying ”Open the meeting book Furniture 1”, ”Next page”, and ”Reset”.

In terms of natural language understanding, the flexible multimodal interaction paradigm imposes some important limitations on the expressiveness of natural language. The most important limitation concerns questions that address more than one speaker, topic or argumentative category, for example ”Why did Mary disagree with Greg’s proposal?”. Currently the system only allows for selecting one speaker per search. If a second speaker is selected, then it replaces the previous speaker criterion. From a natural language understanding point of view, it is possible to interpret questions that combine several speakers, as we have seen in chapter 3, but if the same criteria are selected with a menu-based interface, the meaning of selecting two speakers is ambiguous. The user could either want to find all episodes in which Mary or Greg spoke, or episodes in which Mary and Greg spoke. Furthermore, if the speaker criteria are combined with argumentative criteria such as ”disagreement” and ”proposal”, then the possible interpretations increase even more. In theory it would be possible to design a menu-based interface that allows for selecting not only the search criteria but also the appropriate set operators to express exactly the same query by menu-selection as with natural language, but the use of such an interface would become complex and difficult to use. To preserve the flexible multimodality in



EXAMPLES: Pulp Fiction, Representative of the 90s?, Other chairs, Date, Coffee machine, Next steps, Individual space, Shelves and cupboard, Size, Furniture examples, Armrests, The Usual Suspects, Privacy and collaboration, Colleagues, Door, etc.

EXAMPLES: management, concrete, seventh, franc, update, begin, button, labor, innovative, invoice, chair, cluttered, atmosphere, institution, rethink, etc.

EXAMPLES: The Google Culture, Proposed furniture selection, Mirek's presentation, Denis' presentation, Andrei's notes, Andrei's presentation, Interface concept, Agnes' furniture choices, Susan's notes, Agnes' notes, Teacher Workspaces, Agnes' presentation, DIGS article, Martin's notes, Final furniture selection, etc.

#	Topic
1	Absent
2	Agnes' presentation
3	American Beauty
4	Andrei's presentation
5	Anniversary of liberation
6	Appliances
7	Armchair
8	Armrests
9	Arrangement testing
10	Arrangements for the next meeting
11	Article and magazine racks
12	Awards
13	Balance
14	Balcony

#	Keyword
1	academic
2	academy
3	acceptable
4	accommodate
5	accord
6	account
7	accurate
8	act
9	action
10	active
11	activity
12	actor
13	actual
14	addition

Title	Document type
Agenda	article
Agnes' furniture choices	doc
Agnes' notes	notes
Agnes' presentation	posters
Andrei's furniture choices	slides
Andrei's notes	
Andrei's presentation	
Components design	
DIGS article	
Denis' furniture choices	
Denis' notes	
Denis' presentation	
Final furniture selection	
Interface concept	
...more...	

Figure 4.6: Hierarchical access to content criteria menus

Archivus, a trade-off has been made between expressiveness of natural language and usability of tactile manipulation.

Other natural language phenomena that have been excluded in Archivus are:

- Understanding of terms and expressions that refer to more general concepts than the ones in the system, for example "meetings in *Switzerland*" which specifies a country instead of a city, or "*reactions* to the proposal" which includes both agreements and disagreements.
- Understanding of fuzzy terms for dates, for example "past three months" which represents three different values for the criterion "month".
- Negation, for example "meetings that Greg did *not* attend" which represents meetings where the value of the speaker criterion can be anything except "Greg".

Some natural language understanding that has been included even though it does not have the precise equivalence in tactile manipulation is:

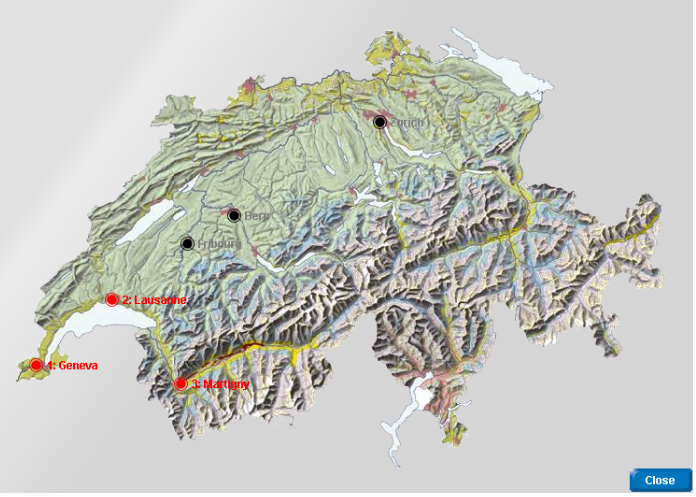


Figure 4.7: Map-based representation of location criteria

- Synonym handling of terms and expressions that are close to the concepts in the system but not entirely synonyms, for example "reject" and "be opposed to" which are recognized as "disagreement".

- Contextualization of spoken system responses based on how a request is formulated.

An example of the last point is when a user wants to find out where a meeting took place, for example the movie club meeting. The information can be found in at least two ways: either the user can express the command "Open the movie club meeting", or ask "Where did the Movie club meeting take place?" In both scenarios the system opens the cover page which reveals the meeting location. But in the first case, the system responds with the same prompt as when the book is opened with tactile manipulation: "What would you like to find in this meeting?", whereas in the second case, the system responds to the question with an appropriate answer, for example "The requested information is displayed."

It is important to note that the natural language understanding in Archivus is not fully implemented, as the objective of this work is to first elicit requirements for how to implement it. During evaluations with users the interpretation of natural language input is simulated by a human 'wizard' who ensures that the natural language understanding performance is maximized and that the system responses are appropriate, while at the same time maintaining consistent system behaviour. The evaluation method is described in the next section.

4.3 The Wizard of Oz evaluation method

4.3.1 Wizard of Oz for speech interfaces

Evaluation of natural language interfaces differs in a crucial way from evaluation of graphical user interfaces. Natural language interfaces are transparent, i.e. they typically do not reveal all the possible interactions that a user can make at a given system state. One way to learn the possible interactions is by making a request and observing the reaction of the system. If the user interacts with a fully implemented system, they quickly learn the *limitations* of the system and interact based on these. So if the goal of the evaluation is to elicit requirements for improving the system, for example to extend its functionality or add new words to the language vocabulary, the natural language interactions are not representative of the input that the future system should be able to deal with. In order to gather representative data, the user needs to interact with a system that behaves like the future system. But implementing such a system to work flawlessly without intermediate evaluations is impossible, since one cannot foresee how a user wants to interact with it.

The solution to this problem is to let the user believe that the system is fully implemented when in fact a human wizard simulates those components that are not yet implemented. This evaluation method is known as the Wizard of Oz (WOz) method (Dahlback *et al.*, 1993). Originally WOz was developed for speech interfaces, for example telephony applications. In such applications, the user talks to the system over a phone line, and the wizard simulates some or all of the natural language processing components, including speech recognition, natural language understanding and response generation (see figure 4.8).

The WOz method was chosen for evaluating the Archivus system, because one of the objectives of the evaluation is to gather representative data for developing the natural language understanding component. The backend dialogue system in Archivus is developed with the Rapid Dialogue Prototyping Method (RDPM) for speech interfaces (Melichar *et al.*, 2006). In this prototyping method, the wizard's control interface, which is used for simulating missing components of the system, is an integrated part of the system design. Therefore, it is technically possible to perform WOz evaluations with Archivus without making additional effort to develop the evaluation framework. In practice, however, there are important differences between speech interfaces and multimodal interfaces. The default WOz framework for speech interfaces needs to be extended considerably in order to enable evaluation of multimodal graphical user interfaces. The next section describes these extensions.

4.3.2 Multimodal Wizard of Oz

The main difference between WOz evaluations for voice-only applications and multimodal applications is that the hardware and software configurations for the first are much simpler. In case of telephony applications, the user interface and wizard's control interface can run on a single computer. Only the audio-signal needs to be transmitted to the user. In fact, there is no requirement to implement an actual system to run a WOz evaluation. The wizard can

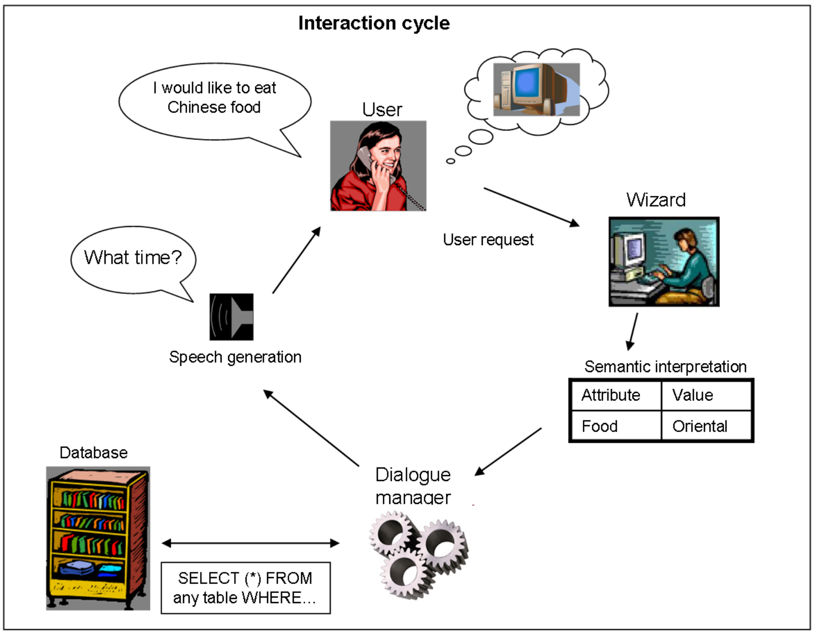


Figure 4.8: The Wizard of Oz evaluation method for speech interfaces

simulate all components of the system if necessary. Multimodal systems that have graphical user interfaces impose more constraints on the setup of WOz evaluations. Instead of one computer, at least two are required for running the system. The first displays the graphical user interface to the user, and the second has the control interface for simulating the missing system components. The two computers need to be located in different rooms and connected via a network so that the user and the wizard both can manipulate the system.

Another constraint imposed by graphical user interfaces is that the wizard is limited in its actions by the existing automated components, in particular if the system is designed to be flexibly multimodal, as in the case of Archivus. When the wizard receives language input, the possible actions that he or she can perform are strictly defined by the possible actions that the user can perform with tactile manipulations in the graphical user interface. This does not mean that the wizard's monitoring task is easier than in unconstrained WOz experiments with voice-only systems. On the contrary, the wizard now has to monitor and control many more components and functionalities of the system. As opposed to voice-only systems, the wizard is not only concerned with audio-input, but has to coordinate input coming from different modalities simultaneously. The interpretation of this input may highly depend on what is shown on the user's screen. Also the natural language output generation needs to be coordinated with the graphical output. To accomplish this complex management of tasks, the wizard needs to have a view of the user's screen to be able to follow the changes in the graphical user interface and to take these into account when processing the input. In addition, the wizard may also want to see the video of the user to better understand their actions and responses. Finally, to be able to react quickly to the input and to manipulate the system efficiently and consistently, the control interface must be appropriately designed for the task. This is achieved by running pilot experiments and evaluating inefficiencies in the wizard's operations. We go into more detail on this issue in section 4.3.5.

From the user's point of view there is also an important difference between natural language interfaces and multimodal interfaces, which influences the WOz evaluation setup. In voice-only applications users tend to be more tolerant to failures and slow system response time. The novelty of speech compensates for system inefficiency (Rajman *et al.*, 2006). As a result, the wizard can spend up to 5 seconds processing an input without raising suspicion or frustration in users. In multimodal systems however, the user is partially interacting through tactile manipulation, and such manipulations are expected to be processed immediately by the system. Even if the wizard does not need to monitor tactile interactions (in most cases their processing can be fully automated), the combination of natural language and tactile manipulation in the same interface raises the expectation on fast system response time also for natural language input. If the language processing is too slow, it risks causing harm to the evaluation in two ways: 1) the illusion of an automated system is broken, or 2) that the user becomes too impatient and chooses tactile interaction instead of natural language. For these two reasons it is crucial that the wizard is able to operate very fast and consistently for every received natural language input.

The problem of achieving fast and consistent simulation of language processing is that the wizard's cognitive load is much higher in multimodal WOz simulation than in traditional WOz. There is simply a limit to how much a human mind can process at a given time. However, there are ways in which the cognitive load of the wizard can be reduced. One solution, suggested by Salber and Coutaz (1993) is to introduce a second wizard into the simulation and let each wizard specialize in different tasks. With this approach each wizard can focus on a smaller set of operations and react more quickly to user input. Another solution is to allocate time and effort into the design and ergonomics of the wizard's control interface, to facilitate the wizard's cognitive tasks when processing the input. For example, for some specific operations it can be possible to reduce the number of steps the wizard needs to go through, or when an action involves browsing a list, it can be possible to reduce the amount of items to search through. Such small improvements can make a vast difference in wizard's cognitive load and consequently in their performance.

Extending the WOz methodology to multimodal interface evaluations has represented a considerable part of the work on setting up and running evaluations with Archivus . In the remainder of this section, we will describe the technical setup and the wizard's control interfaces that were developed through several sets of pilot studies.

4.3.3 The user's environment

In the Archivus Wizard of Oz environment the user sits at a standard desktop PC with a 15 inch screen and a wireless mouse and keyboard. A lapel microphone is pinned to the user's shirt to register speech input. The user is therefore able to provide input in three modalities. The graphical user interface in turn gives feedback in three modalities: graphical and textual feedback using the screen, and spoken feedback using audio speakers.

The user's actions are recorded by two cameras situated on tripods (see figure 4.9). One camera records the face, and also transmits a live video stream to the wizard's room. The other is placed on the side, slightly behind the user, to record interactions done with keyboard and mouse, but also to film the user reading the manual, writing down answers to questions and other activities relevant to the evaluation. The desktop screen and audio from lapel-microphones are recorded on a third video. All in all, the experiments are recorded from three parallel views for post-evaluation analysis. Additional detail on the experimental setup can be found in (Rajman *et al.*, 2006) and (Lisowska *et al.*, 2009)

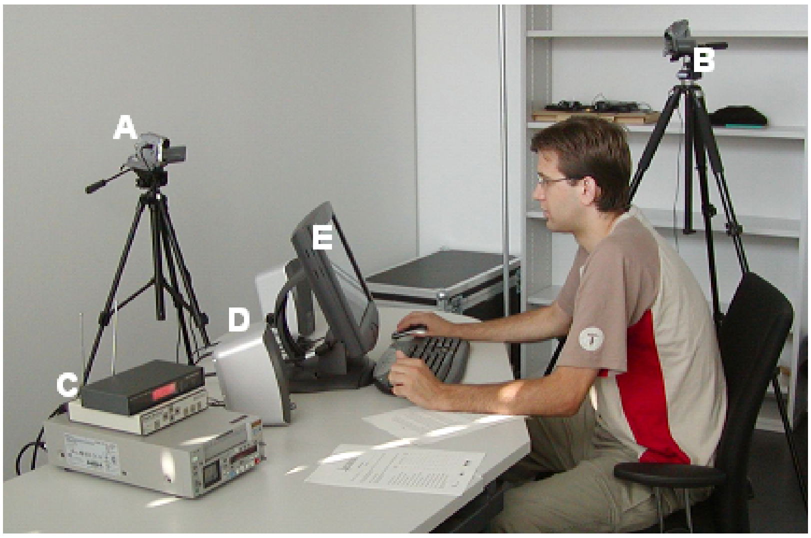


Figure 4.9: View of the user's work environment: (A) Camera that records and streams the user's face to the wizards' room (B) Camera that records the user's hands (C) Equipment for recording the user's screen (D) Loudspeakers (E) Multimodal user interface

4.3.4 The wizards' environment

To maintain the illusion of an automated system, the wizards work in a different room than the user. This room is equipped with two monitors that provide views of the user's screen and the user's face (see figure 4.10). The second also plays the audio from the user's lapel microphone. This configuration gives the wizards a sufficient overview of the situation in the user's room to be able to determine when to react and how to act upon received input. The wizards see where the user's attention is focused, what the user can see on the screen, what they do with the interface, and how they react to system responses. The face view is also used for detecting unexpected situations that could potentially influence the experiment - for example the user blatantly ignoring the instructions given by the experimenter, the mouse not working due to discharged batteries, the user sending an SMS during interaction, or technical and cleaning personnel entering the room during the experiment (these are all examples of real situations we experienced during our WOz experiments).

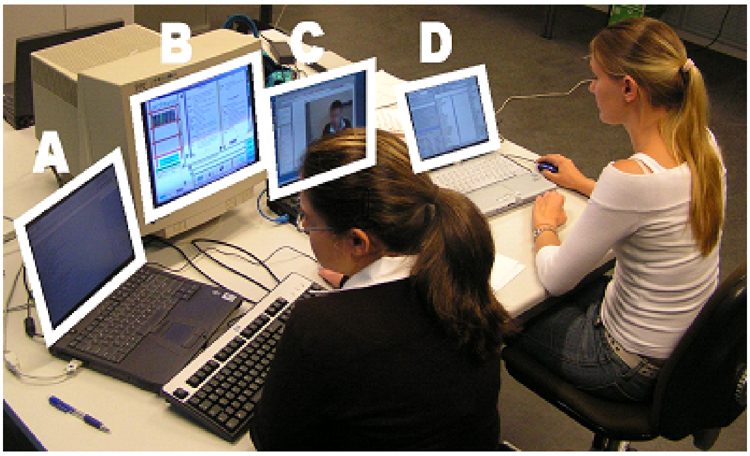
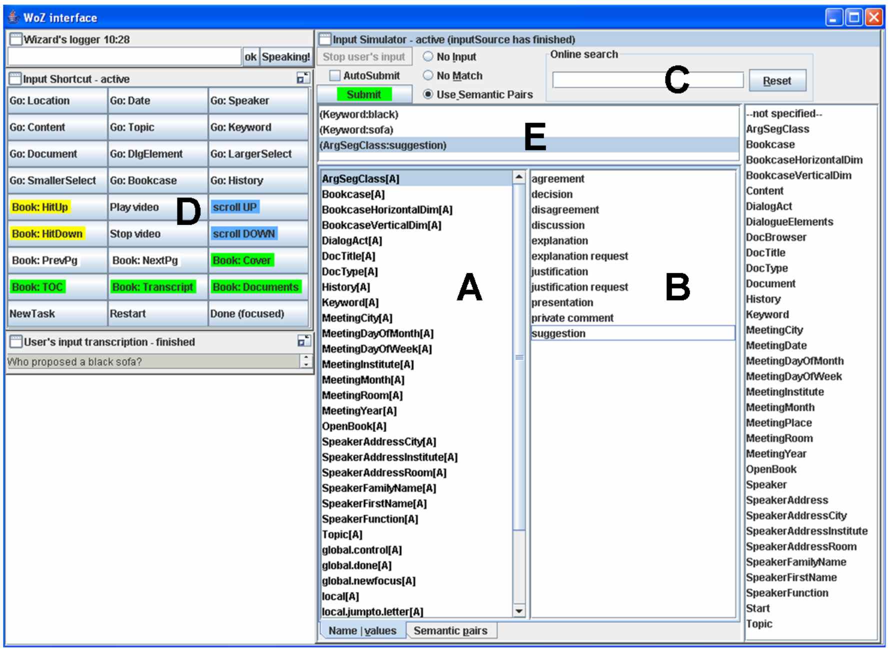


Figure 4.10: View of the wizard's environment: (A) User's screen (B) User's face (C) Input control interface (D) Output control interface

The two wizards that work in the room each have their own laptop with a control interface specially designed for their task. The first wizard is responsible for the input processing and has a control interface for monitoring and generating attribute-value pairs from the user's input. The second wizard is responsible for output generation and has a control interface for supervising and, if necessary, replacing default prompts that are spoken by the system. In the early stages of setting up the WOz environment for Archivus, the Wizard's setup consisted of only the input control interface that was automatically generated with the Rapid Dialogue Prototyping Methodology (Melichar *et al.*, 2006). The output generation was fully automated and the wizard did not have tools for changing default prompts. Our initial assumption was that the default prompts did not need to be supervised. However, during pilot experiments it was discovered that the frame-based dialogue model in some situations was too simple for the graphical user interface and that system prompts were not sufficiently contextualized. A second control interface was developed to enable the wizard to manipulate system prompts. At this stage we decided to introduce a second wizard, as it was already evident that the cognitive load of the first wizard was too high to take on additional tasks, and it was a natural way to divide the work into input and output monitoring.

4.3.5 The input wizard's control task

The input wizard is responsible for supervising input coming from spoken and written natural language and mouse pointing. The task consists in translating the input into attribute-value pairs that can then be processed by the dialogue manager. If the user clicks with the mouse, the pairs are generated automatically. If the user types, the system generates a default interpretation and the wizard checks its correctness and, if necessary, modifies it. The most demanding task is to simulate spoken input. Then the wizard has to select the appropriate pairs from among the available ones in the system. The available pairs are displayed category-wise in scrollable lists of different sizes (see figure 4.11)



Who proposed a black sofa?

(Keyword:black)
(Keyword:sofa)
(ArgSegClass:suggestion)

Figure 4.11: Input Wizard's control interface: (A) Database attributes (B) Database values (C) Quick search for attributes and values (D) Short-cut buttons for GUI commands (E) Interpretation of user input

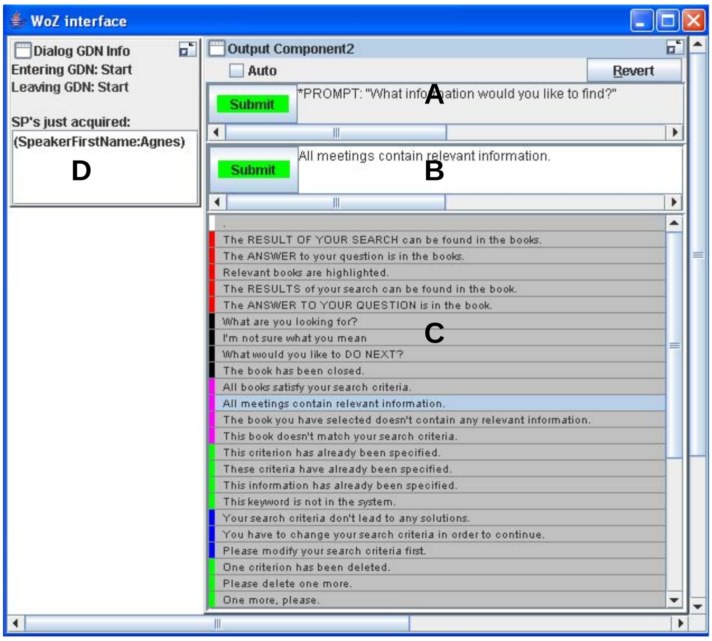
There are two types of attribute-value pairs in the system. The first kind are the data pairs, also referred to as search criteria. Users specify these when searching in the meeting data. For example, when a user asks: " *Which meetings happened on April 21st?*", the wizard translates the request into the two attribute-value pairs *Month:april, DayofMonth:21.* For some attributes, such as meeting topics, there are hundreds or even thousands of entries in the database. Although they are alphabetically ordered in lists, the wizard loses valuable time and cognitive effort on scrolling down and searching for a value that they already know they want to select. Therefore, one of our first improvements in the input control interface was to add a quick-search box that allows the wizard to filter among the database entries. (See figure 4.11 (c))

The second kind of attribute-value pairs in the system are the GUI pairs. Users specify these when they want to manipulate the graphical user interface, for example to open a book (*Bookcase:Furniture1*), change page (*OpenBook:nextPage*) or reset the system (*Global:restart*). In most cases, users prefer the mouse for such interactions, which means that the wizard does not need to simulate anything. But frequently users perform these actions with voice commands, and then the wizard has to act very fast to be able to compete with the speed of mouse clicks. In the original design of the input wizard's control interface, the wizard selected the GUI pairs in the same manner as the data pairs, i.e. by scrolling lists. However, we discovered that it was cognitively much more challenging for the wizard to memorize the different categories of attribute-value pairs for GUI navigation than for data retrieval, so the wizard often lost time searching for the appropriate list of values in their control interface. Even when the wizard found the proper list immediately, it meant several interaction steps to select the list, then the value, and finally to submit the pair to the dialogue manager. To overcome this inefficiency, we added a set of short-cut buttons into the control interface that allowed the wizard to perform the action in one step instead of three (see figure 4.11) We decided to include only the most frequent voice-commands as it would take too much space to create one button for every possible GUI navigation action in the system, and it would not make the wizard more efficient if the interface would be too cluttered with buttons. This trade-off was found to work well in practice.

4.3.6 The output wizard's control task

The second wizard is responsible for monitoring the spoken feedback of the system. As previously mentioned, it was not foreseen in the original setup that this second wizard would need to change the default prompts that were generated by the dialogue manager. However, during pilot experiments we identified situations in which the default prompts were not optimal, sometimes even directly misleading. For example, there were situations in which a user had specified too many search criteria and arrived at a "dead end" with no results. Many users spontaneously tried to get out of this situation by asking a new question, which was not the right approach because it would only lead to more search criteria and an even stronger dead end. The problem with the default system prompt occurred when a meeting book was open at the time of the dead end. Then the system would just keep repeating "This book doesn't contain the information you are looking for" (this is the default prompt for all situations in which a user opens a book that does not match the current search criteria). The user understood the feedback as if all the questions they asked were "wrong" questions about that specific meeting. A better prompt in this situation would be for example "You have too many criteria. Please remove some first or start a new search".

To avoid such confusing and badly contextualized prompts as in the example above, the second wizard's task is to determine after each user interaction if the proposed system prompt is appropriate or not. In most cases it is, and then the wizard only needs to confirm the prompt, which causes a delay of approximately 1 second compared to fully automated prompt generation. If the prompt is not ideal, the wizard either types in a new one manually, which is very infrequent (0.3% of the entire output wizard's actions during evaluation) or selects a pre-defined prompt from a list that stores all the previously typed prompts. (see figure 4.12) The last takes approximately the same amount of time as confirming default prompts. The reason is that the wizard can predict which prompt to select as soon as the user provides their input, and while the input wizard is processing that input, the output wizard can prepare the feedback simultaneously, thereby saving system response time.



Entering GDN: Start
Leaving GDN: Start

SP's just acquired:
(SpeakerFirstName:Agnes)

*PROMPT: "What information would you like to find?"

All meetings contain relevant information.

The RESULT OF YOUR SEARCH can be found in the books.
The ANSWER to your question is in the books.
Relevant books are highlighted.
The RESULTS of your search can be found in the book.
The ANSWER TO YOUR QUESTION is in the book.
What are you looking for?
I'm not sure what you mean
What would you like to DO NEXT?
The book has been closed.
All books satisfy your search criteria.
All meetings contain relevant information.
The book you have selected doesn't contain any relevant information.
This book doesn't match your search criteria.
This criterion has already been specified.
These criteria have already been specified.
This information has already been specified.
This keyword is not in the system.
Your search criteria don't lead to any solutions.
You have to change your search criteria in order to continue.
Please modify your search criteria first.
One criterion has been deleted.
Please delete one more.
One more, please.

Figure 4.12: Output wizard's control interface: (A) Default system prompt (B) Field for editing system prompt (C) Wizard's pre-defined prompts (D) Contextual information about the last user input

4.4 Research questions and hypotheses

The objective of setting up a complex multimodal Wizard of Oz framework, as described in 4.3, is to obtain an evaluation framework in which we can study how users want to interact with a meeting data retrieval system when they are free of fundamental limitations in natural language processing performance. The goal is to incent users to speak freely and express in their own terms what they want to find in meeting data. Tactile interactions are intended to be used only when they are superior in efficiency or practicality. Having created such an evaluation framework, the research questions we want to address are divided into two main objectives. The first one is to collect real user queries about meeting discussions, as opposed to survey-collected ones, and the second is to assess the importance of natural language as opposed to standard GUIs for searching in this type of data.

4.4.1 Natural language querying of meeting data

The first question of interest is whether users ask the type of questions that were collected in the user requirement analysis described in chapter 3. This issue is important because it gives us a new, possibly more realistic estimation of the amount of work required to develop meeting data retrieval systems in this domain. More precisely, the first research question that interests us is:

> **R1**: What type of information do questions made to a transcript-based meeting data retrieval system pertain to?

This question addresses whether it is worth the effort to annotate meeting discussions with higher-level discourse annotations to improve retrieval. In particular, we want to evaluate the usefulness of the argumentative annotation, for which there are not yet any standardized, formal annotation schemas. In short, we ask if the current progress in argumentative annotation of meeting discussion is promising for this domain.

To answer this question, it is not necessary to evaluate a natural language interface to meeting data. Any interface that allows for searching or browsing argumentative annotations will do. The central issue is whether users choose to access the argumentative annotation in their search and whether those annotations help users to retrieve the relevant episodes of meeting discussion. As Archivus is a multimodal interface there are two possible ways of exploiting argumentative annotations. The first is to ask questions in natural language about the argumentative aspects of meeting discussion, for example "Was there disagreement about the colour scheme?". The second is to select the argumentative criteria from a list, in this example the value "disagreement". Intuitively it seems more natural to choose natural language for expressing requests about argumentation, because such requests often involve other interrelated search criteria. All these criteria can be expressed easily as a single natural language query. For example, in the example above, the query contains an argumentative criterion (disagreement) and a topical one (colour scheme). In practice we need to take into account that the subjects in our evaluations have no prior experience of meeting data retrieval systems or argumentative structuring of meeting discussions. This means that they do not know the possible argumentative categories that they can specify as search criteria. Previous studies have shown that users are uncomfortable with using natural language when they do not know the precise set of commands available in the system (Sears and Jacko, 2007). From this outset, it seems more likely that users will prefer to explore the annotations with tactile selection instead. Once they are familiar with the argumentative categories in the system, they may switch to natural language queries. The reason why the modality choice is important in the Archivus system is that it may have an impact on how frequently the argumentative criteria are chosen for search. With list selection the user is forced to choose an order in which to select criteria. Our hypothesis is as follows:

> **H1**: When users select criteria from menus, they prioritize more familiar types of criteria first, such as keywords and topics. Argumentative criteria are chosen as an additional search feature only when content search alone fails.

In other words, for many searches the argumentative annotation will not be exploited because content criteria will be sufficient, if not as efficient. On the other hand, if the criteria are selected with natural language, our hypothesis is the following:

> **H2**: When users select criteria in natural language the argumentative annotations are exploited as often as they are relevant to the search, because users do not need to choose their criteria in order, but can specify all of them in one query.

By comparing a linguistic version of Archivus where users can choose between NL query and menu-selection, and a non-linguistic version where users are forced to select from menus, we want to show that natural language querying is more adequate for exploiting argumentative annotations than menu-based selection, and that users who exploit these annotations are more efficient in retrieving relevant episodes of meeting discussions than users who do not.

The second research question concerns the natural language understanding of questions:

> **R2**: What level of linguistic analysis is required in order to correctly interpret questions in the domain of meeting data retrieval?

To answer this question, we need to analyze the linguistic surface properties of the questions that are made to the system. This issue is important because the precision and recall of interpreting questions is in fact closely related to the usefulness of annotating meeting discussions. If the precision is not perfect, some references to meeting concepts present in the query may be incorrectly interpreted and lead to irrelevant search results. Such errors could strongly influence the usefulness of various annotations made on meeting discussions. If the recall is not perfect, some of the meeting concepts in the query may not be extracted during interpretation and lead to incomplete search results. Such errors would prevent users from exploiting the annotations during search. As we showed in chapter 3, the syntactic diversity of terms and phrases for referring to topics and argumentation is rich, and simple template-based approaches are not capable of detecting and disambiguating all the possible meanings, whereas semantic role extraction and few generic interpretation rules do the job with very high precision and recall. The hypothesis that emerged from the work in chapter 3 is as follows:

> **H1**: The natural language understanding of questions requires domain-specific word sense disambiguation and extraction of semantic roles from the sub constituents of the sentence in order to detect and disambiguate all references to concepts and relations of a formal model of meeting discussion.

In the context of a multimodal system, however, we believe that the fact that users have the option to choose criteria from menus has an influence on the way they express queries also in natural language. The user is guided into thinking of the search in terms of attributes and values rather than a natural language query engine. Therefore, a second more adapted hypothesis is:

> **H2**: Real questions to a flexibly multimodal meeting data retrieval system are syntactically short and simple compared to theoretical questions given as examples to an imagined, future system.

If users do not ask sentence-like questions that have linguistic structure, advanced word sense disambiguation and semantic role extraction may not be applicable to the natural language input. In that case, the heuristics of a template-based approach may be the second best approach. Another option is to design an interactive language understanding component where the user disambiguates the possible interpretations by hand. Such methods have been proposed in other domains (Nakao *et al.*, 2006) and have the advantage of giving the user full control of how their input is interpreted. In such systems there is no 'hidden' language understanding involved. However, an obvious drawback is that the disambiguation leads to less natural interaction and may not differ substantially from a non-linguistic menu-based approach for selecting search criteria.

4.4.2 Natural language interaction in a graphical user interface

The second objective of performing multimodal Wizard of Oz evaluations with the Archivus system is to learn to what extent natural language as an interaction modality contributes to more efficient meeting data retrieval. The research question we want to answer is:

> **R3**: What are the properties of natural language that motivate the integration of natural language into meeting data retrieval interfaces?

This question assesses the value of a natural language interface to meeting data, as opposed to a standard graphical user interface. Natural language interfaces are more costly to develop, so it is indeed relevant to specify in what specific circumstances they are needed, before making the effort to develop a fully automated system. The fundamental assumption here is that both natural language and graphical user interfaces are useful for meeting data retrieval and that integrating them both into one unified interface is more useful than developing one or both as stand-alone interfaces. The design of the Archivus system as a flexible multimodal user interface allows us to compare in a very precise manner which actions users prefer to perform in natural language and which ones they prefer to perform with tactile manipulation.

On a very coarse level, the possible interactions with the Archivus system can be divided into search actions and browsing actions. The search actions consist of all actions that generate or change the active search criteria. To generate criteria the user can either express natural language queries freely or access the different categories of search criteria buttons that display the possible attributes and values for that category. To search in the meeting data in Archivus, there are five categories and in total 19 attributes to choose from as search criteria. Some of these attributes are easy to distinguish, such as the attributes for speakers and dates. Some on the other hand are less intuitive. Two examples are the distinctions between the topic and keyword attributes, and the novel attributes argumentative categories and dialogue acts. It seems to require less cognitive effort to express such search criteria in natural language and to let the system handle the assignment of attributes and values, than to learn which attributes belong to which categories and then to choose manually among them. The first hypothesis is therefore:

> **H1**: Natural language querying is more efficient for search than tactile menu-selection when the user is not sure which precise attributes and values that best reflect what they are searching for, and when it requires additional cognitive effort to choose the right attributes.

Studies on natural language interfaces to databases have also addressed the issue of selecting attributes and values with menus versus natural language. Walker and Whittaker (1989) reported that nave users who are not familiar with the precise structure of the database appreciate natural language as it allows them to refer to the attributes (or in this case database fields) with their own vocabulary. The cognitive effort of learning and memorizing attributes is in this case removed, because users refer to the data based on their previous existing knowledge of the domain rather than the structure of the database. Only natural language can enable users to do this.

Another aspect of search that is relevant to determine the usefulness of natural language querying is the scope of information that is stored in the system's database. In the case of meeting data, for example the scope of meeting dates is limited by the days of the calendar. Moreover, the actual instances of meeting dates in the database are limited to those for which a meeting was recorded. If the user wants to search for meetings that happened during given periods, it seems more efficient to select date criteria from a calendar style GUI element that shows the available dates, than to ask for dates in natural language and possibly get no results because no meetings happened at those dates. Dates are an example of a limited-scope category of search criteria. Also meeting locations and participants belong to such search criteria. For all these, it is possible to think of a GUI design in which tactile selection is more efficient than natural language querying. For locations it may be a map, for meeting participants for example photos. The main argument is that such criteria do not need to be presented solely as menus. Other, more efficient selection methods are possible. However, there are other types of search criteria for which there are no obvious ways of displaying them visually because the scope of possible values is either very large or has no visual mapping. In the meeting data, topics and keywords are good examples, but also discourse criteria belong to this type. If the user knows approximately what value to specify for their search, it seems more efficient and effortless to express the request in natural language than to scroll down long menus. A second hypothesis hence follows:

> **H2**: Natural language is more efficient for search than menu-selection when the range of possible values is unrestricted, and the user knows which value they want to select for their task

The next part of possible interactions in the Archivus system concerns browsing. The browsing actions are all the actions that involve access and review of the meeting data. Typical examples are opening a book, changing page and playing the meeting recording. There are many arguments for why natural language is not the most appropriate modality for performing these actions. Users may not know how to refer to GUI elements in the interface if they are not labelled with names. Browsing pages with mouse clicks is faster than speaking a command and waiting for the natural language processing to finish, and it may even be tedious to repeat the same voice command over and over (e.g. ”next page”, ”next page”,). If the meeting recording is being played, the audio interferes with the voice commands and leads to speech recognition error, even when a human wizard is simulating it. In spite of all these drawbacks, natural language does have a property that gives it an advantage towards tactile manipulation. Referring to GUI elements in natural language does not require locating them physically on the screen first. For novel users in particular this property can be very useful as it can help them to navigate to GUI elements that are initially hidden, such as the referenced meeting documents in the book appendices. Another advantage of speech is that it can be used synchronously with other non-linguistic actions. Two hypotheses derived from this reasoning are:

> **H3**: Natural language makes GUI navigation more efficient by allowing short-cuts to GUI elements that are not visible on the screen

> **H4**: Speech contributes to more efficient browsing when the user's hands are busy, for example making notes with a pen

The hands-free property of natural language has been listed as one of the most important speech functionalities in formal modality theory (Bernsen and Luz, 1999), and speech interfaces have been successfully implemented for many hands-busy tasks such as car driving (Cox *et al.*, 2008). In the context of office environments where computers are typically desktops or laptops, there are naturally less situations in which the hands are busy while interacting with the computer, but also in this context we believe that the hands-free property can be useful, even if not strictly necessary. In a more general perspective, natural language can enhance the experience of using the system, even if it does not contribute to more efficient interaction in any measurable way, for example in terms of speed or task completion. User satisfaction is an important factor that determines users' willingness to use a system again in the future. Natural language interaction could be an aspect that increases user satisfaction. The last hypothesis thus is:

> **H5**: Natural language contributes to higher satisfaction with the general functionality of a meeting data retrieval system, even when it does not lead to more efficient retrieval

In the literature there are reports that relate to the above hypothesis, for example that users appreciate a more human-like natural language-based interface once they have experienced it (Qvarfordt *et al.*, 2003). With our multimodal WOz evaluation of the Archivus system we want to complement the research by showing that meeting data retrieval is potentially one of those domains in which natural language interaction with a graphical user interface is appreciated by users.

4.5 Experimental method

To be able to answer the research questions outlined in 4.4, we designed an experiment in which users were told to search for information in recorded meetings using the Archivus system. More specifically, we designed a set of experimental conditions in which users had access to different interaction modalities (see section 4.5.1). We entered a set of recorded meetings into the database and edited the data to comply with our needs (section 4.5.2). The task was defined as a set of questions which were based on the available meetings in the database (section 4.5.3), and the experimental procedure with all required documents, such as tutorial and questionnaires, were created (4.5.4). When the experiments were executed, the interactions were recorded on video and logged by the system. This data was then post-processed to enable us to perform various types of analysis to test the hypotheses that we have formulated (4.5.5).

4.5.1 Evaluation conditions

Several versions of the system were developed for the experiment, to be able to compare users based on their access to different interaction modalities. These evaluation conditions were defined with several research goals in mind, including some that are outside the scope of this thesis (Lisowska, 2007; Melichar, 2008). There were 10 evaluation conditions in the experiment that corresponded to 10 combinations of the modalities voice (V), keyboard (K), mouse (M) and pointing on touch-screen (P) (see table 4.1).

Single modality	**Two modalities**	**Three modalities**
V	VK	MVK
P	PV PK	
M	MK MV	PVK

(V: Voice, K: Keyboard, M:Mouse, P:pen)

Table 4.1: Evaluation conditions designed to give users access to different subsets of interaction modalities in the Archivus system

In this thesis, we are mainly interested in comparing non-linguistic conditions with linguistic ones. Some conditions are almost equivalent according to this distinction. For example the MV and PV conditions both offer a linguistic and tactile interaction modality. Since this thesis does not address the differences between different tactile modalities, such as in this case mouse and touch-screen pointing, our analysis will be limited to conditions that use mouse. In the case of linguistic conditions, we analyze both speech and keyboard conditions as there may be differences in the type of input that users provide through these two modalities. These differences may influence the outcome of the analysis in perspective of the various research hypotheses being tested. The evaluation conditions that are most interesting in this study are the M, V, VK, MV and MVK conditions.

4.5.2 The meetings in the database

For our experiment we chose to include six recorded meetings in the Archivus database. The meetings are not 'real' meetings but acted by the participants based on a predefined meeting topic and speaker roles. The reason why the meetings are simulated is that they needed to be recorded in a Smart meeting room (Nijholt *et al.*, 2006) with special recording equipment where the audio and video signals of each speaker were recorded into separate channels, to enable advanced data post-processing. The Smart Meeting room and the meeting recording were done as part of the Swiss Interactive Multimodal Information Management (IM2) project [1].

The six meetings consist of one design meeting where the goal is to develop an innovative remote control, a series of four meetings where the participants choose furniture for a reading room, and one movie club meeting where a film is chosen for the next movie club screening. These meetings were chosen because they represent discussions where argumentation and decision-making processes are central to the meeting. Other types of meetings where the goal is, for example, to report progress in a project are of less interest to the current research.

Most of the post-processing and annotation of the recordings was already available when the meetings were entered into the database. Specifically, the meetings had been transcribed and segmented into utterances. Each utterance had been assigned with the identification of the speaker and dialogue acts, e.g. 'statement', 'question' and 'positive answer' (Popescu-Belis *et al.*, 2004). The meeting discussion had also been segmented into topics (Georgescul *et al.*, 2005) and manually annotated with topic labels. There are different ways of labelling a topic, for example by extracting the most frequent keywords in the segment and using lexical resources to select the most appropriate generalization of these keywords. In this experiment the topics were written as a mix of words and descriptive phrases, e.g. "Colour" or "Purpose of the room". The goal was to be able to use the topics for two distinct purposes in the system. The first was to make them available as search criteria. The second was to extract a table of contents of the meeting to facilitate browsing of the meeting book.

[1] http://www.im2.ch

The work that was left to do on the meeting data was the argumentative annotation. A simplified argumentation schema was developed to ensure that non-experts (in this case the subjects of the experiment) would be able to understand and use all of the categories for searching and browsing meeting discussions. The simplification was also motivated by the difficulty of the annotation task. Two annotators were responsible for annotating three meetings each and reviewing the annotations of the other annotator. The simplified annotation schema consisted of 8 argumentative categories that were used for flat annotation of argumentative contributions without sub categories or links to other contributions. The 8 categories were: suggestion (previously propose), agreement, disagreement, explanation request, explanation, justification request, justification, and decision.

4.5.3 Task

Users were told to imagine themselves in a scenario where they were new in a company and had been asked by their superior to find some facts about past meetings using the Archivus system. The task was to find answers to predefined questions by searching in the six meetings that were in the system's database. The questions were formulated in two ways: either as true-false statements or as short-answer questions. In perspective of the two research objectives, namely to elicit queries and to study natural language interaction in a multimodal interface, the questions were chosen in such a way that they covered aspects of both querying and GUI navigation. More precisely, the task was not limited to finding episodes of meeting discussions but also to find general information about the meetings, or find specific pieces of information from the referenced documents. There were in total 40 questions in the task, which were classified into five types according to the information that they pertained to (see table 4.2)

During post-evaluation analysis the answers to the questions were scored not only based on whether they were right or wrong but also based on how close a user was to answering the question. Most subjects were not native English speakers, so it was important to take into account that wrong answers could have been caused by something else than the failure to retrieve the relevant information from the system. The user could have misunderstood the question or been unfamiliar with words and expressions in the meeting discussion that were crucial for answering the question. The scoring was done on a four point scale, as described in table 4.3.

4.5.4 Procedure

The experiment took 2 hours per subject and was divided into four parts. In the first part, the user was given 20 minutes to fill in a demographic questionnaire, read the evaluation scenario, and learn to use the system with a step-by-step tutorial. The tutorial was designed to demonstrate the functionalities of the system and the use of the interaction modalities without biasing the user concerning when to use which modality and how (Lisowska *et al.*, 2007). The tutorial hence contained several search examples where both linguistic and non-linguistic approaches were used. The user was also asked to sign a consent

	Question type	Example
1	The goal is to navigate to the appropriate information in the GUI. No search criteria are needed to find the information.	Where was the design meeting held?
2	The information to be found is an episode of a meeting where a given topic was discussed or certain keywords were said	Appliances were discussed in the Furniture 1 meeting
3	The information to be found is an episode in a meeting where given argumentation or decision-making occurred	Which movie did they finally decide to show?
4	The goal is to find information that is related to a participant of a meeting	Which two participants brought PowerPoint presentations to the Movie Club meeting?
5	The goal is to find something in a referenced document	How many pictures are there in the Google document?

Table 4.2: Classification of questions in the task based on the meeting data that they refer to

Score	Motivation
0	The user found irrelevant information and guessed an answer
1	No answer was given
2	The user found the relevant information but gave the wrong answer
3	The user found the relevant information and gave the right answer

Table 4.3: Rules for scoring answers to questions in the task

form that allowed us to record the experiment. We informed the user that the data was only to be used for our own research and that no part of it would be made available to the public without their prior consent.

In the second part the user was provided a version of the Archivus system that had a subset of the interaction modalities available according to an evaluation condition. The user had 20 minutes to solve 20 questions that were written on laminated cards. The number of questions was intentionally higher than the time it took to solve them, so that no user would run out of questions before the session was finished. The user was asked to solve the questions in order, and for each question they finished, to write the answer on the card and put it in a box, and then reset the system with a special "task finished" button on the user interface. This procedure was enforced to be able to analyze the experiment data on a question-by-question basis. The "task finished" button was added as a convenience because it logged the end of a question automatically. The box was chosen because it prevented users from going back to old questions if they accidentally found the answer later during interaction. Users were told that if they thought that finding the answer was taking too long, they could leave the card blank and move on to the next question.

In the third part, the user was again asked to solve 20 questions, but this time having access to all the available modalities, i.e. either mouse or touchscreen, and voice and keyboard. This session was included primarily to study learning effects, i.e. how the continued use of modalities is influenced by which modalities have been available when first learning to use the system. The results can be found in Lisowska (2007). In this thesis the session is also valuable for eliciting natural language queries about meeting discussions, which is our first objective of the evaluation. For studying the natural language properties that enhance the interaction with the graphical user interface, the session does not provide representative data, since learning effects may in fact override users' natural choice and this is precisely the opposite of what we want to study.

In the last part of the evaluation, the user was asked to fill in a post-evaluation questionnaire where they could give their opinions about the system. The questions addressed both general system satisfaction and opinions about specific components. The results of this questionnaire are analyzed in the light of the interaction modalities that were available to the users in the first half of the experiment. But since all users had the opportunity to use all modalities in the second session, positive and negative opinions about the system cannot be strictly associated with the linguistic and non-linguistic approaches that were available for search and browsing. This fact has to be taken into account when interpreting the questionnaire data.

4.5.5 Data capture and post-processing

The generated data from the evaluation consisted of automatically generated system log files and recordings from three parallel views (face view, view of user at the desk, and screen view). The log files captured for example what graphical components the user accessed, which modality they used, and which attribute-value pairs their interactions generated. Since the natural language processing was simulated by a wizard, the speech input had to be manually transcribed from the video. The transcriptions were then added to the log files into the corresponding places where speech input had been time stamped. Speech and keyboard interactions were annotated with the following additional information:

- Whether the natural language input consisted of keywords, or a phrase with linguistic structure
- Whether the natural language input corresponded to a query addressing some information that the user was searching for, or a command that the user said to navigate in the graphical user interface
- Whether the entities that the user referred to in their input were visible on the screen or not

The log files with the additional annotations were formatted and entered into a relational database, where each record corresponded to one user interaction with all the relevant information attached to it, such as the user id, task, modality condition, interpretation of input, previous interaction and system response. The evaluation was done with 90 users, which resulted in 60 hours of recording and a database of 17.000 interaction records. The next section describes the analysis and results obtained from this data.

4.6 Experimental results

In the time that was allocated to solve the task, most users managed to solve 10 questions in the first session and 9 in the second, i.e. approximately half of the questions in the task. In the subsequent data analysis, when we compute results based on what question users were solving, the questions Q11-20 and Q30-40 have been excluded because too few users solved them.

In linguistic modality conditions where users had access to both voice and keyboard, the preference for voice was so strong (90% of all linguistic interactions) that we did not obtain sufficient data to make comparative analysis of voice and keyboard-based querying. In the following data analysis, when we compare linguistic and tactile interaction, the linguistic input is represented by both voice and keyboard input.

4.6.1 Search criteria selection

The first part of the data analysis addresses the question of what type of information users search for in a transcript-based meeting data retrieval system (**R1**). More precisely, we want to find out if a pure menu-based interface incents users to mainly specify familiar search criteria such as topics and keywords (**H1**), and if a linguistic interface on the contrary incents users to exploit more novel types of search criteria such as the argumentative aspects of meeting discussions (**H2**).

Modality condition	Specified search criteria (average)
Voice-only (V)	19.0
Mouse-voice (MV)	15.4
Mouse-only (M)	10.9

Table 4.4: Number of search criteria specified in session 1 (Q1-Q10)

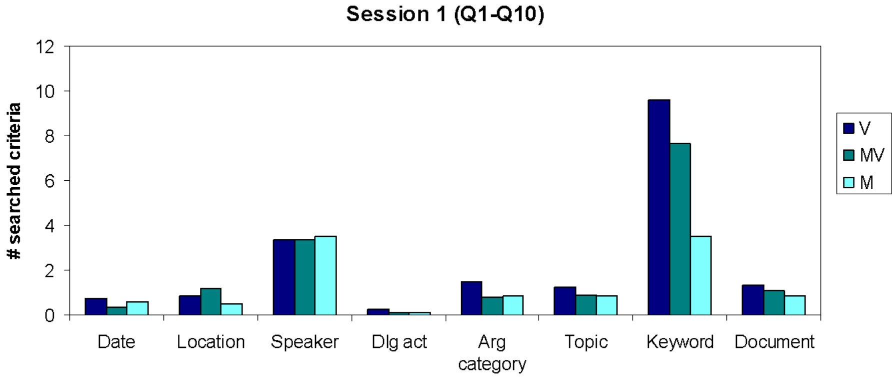


Figure 4.13: Types of search criteria specified during session 1 (Q1-Q10)

On a very coarse-grained level of analysis, we tested if there was a correlation between the user interface and the amount of search criteria that users specified in their task. We compared three interfaces: voice-only (V), mouse-only (M), and combined mouse-voice (MV). We found that users who interacted with the purely linguistic interface expressed more search criteria than users who interacted with mouse alone or with both voice and mouse. (see table 4.4)

In the first session we observed that users who spoke to the system in natural language (V) expressed notably more keyword-, topic- and argumentation criteria than users who selected them from menus with a mouse (see figure 4.13). On the other hand, all users, regardless of interaction modality, chose approximately the same amount of speaker criteria in their search. Overall, keywords (the most familiar criteria) were the most frequently specified during search, whereas novel criteria such as speaker and argumentation criteria were relatively rarely used.

In the second session, when all users had access to voice, mouse and keyboard, the search approaches were even for all users, in particular for keyword search, but users who had started with the voice-only interface continued to choose topic and argumentative criteria more often than other users. (see figure 4.14).

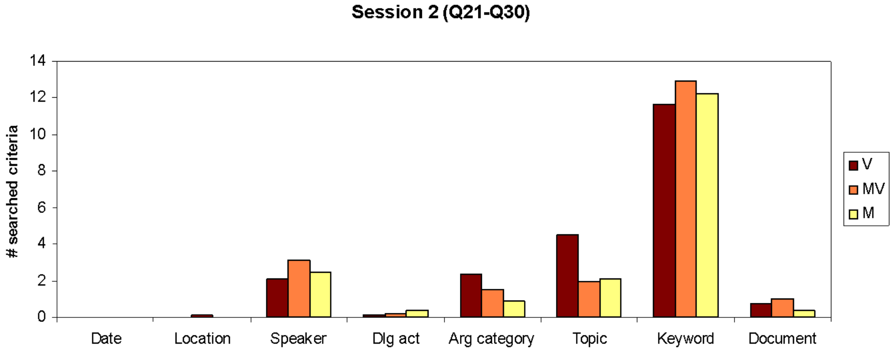


Figure 4.14: Types of search criteria specified during session 2 (Q21-Q30)

On a more fine-grained level, we analyzed the relationship between the questions in the task and the criteria that were specified. For each question in the task we listed the possible search criteria that could be specified to successfully find the answer (see table 4.5), and then used this as a reference when computing the amount of actual criteria that users selected for their search. Questions Q2, Q3, Q5, Q6, Q9, and Q10 are of particular interest to us, as they pertain to the higher-level annotations on the meeting discussions.

In session 1, we first looked at questions that could be answered with content search, i.e. with keyword and topic search (Q3 , Q5 and Q9). We found that users who had a linguistic interface specified more keyword and topic criteria than users who chose criteria from menus (see table 4.6).

For questions where argumentation criteria were called for (Q2, and Q9), most users tried only content search, which was not the foreseen approach for finding the information, whereas very few exploited the argumentation criteria in addition, then mainly users who had a purely linguistic interface. For example in Q2 (”Which two movies does Agnes suggest showing?”) content search was not sufficient for two reasons. The first is that the keyword ”movie” is very imprecise for finding specific episodes of discussion in a movie club meeting. The word occurs repeatedly throughout the whole meeting discussion. The second reason is that when someone suggests a movie, it is not necessarily the case that the person uses the word ”movie” when speaking, for example ”I'd like to propose The American Beauty, because.”. Nevertheless, we found that this keyword was exactly what users chose to search for to answer the question Q2. In the post-evaluation questionnaire, when asked to grade the usefulness of the different search criteria, most users were neutral or sceptical about the usefulness of argumentation criteria but positive about the content and speaker criteria. This raises the question whether users received insufficient support in the tutorial and general design of the system to understand fully how to search with argumentation criteria.

	Cont	**Spkr**	**Arg**	**Doc**	**Date**	**Loc**	**None**
Q1: The Furniture 4 meeting took place on March 10th, 2004.					x		
Q2: Which two movies did Agnes suggest showing?		x	x				
Q3: Appliances were discussed in the Furniture 1 meeting?	x						
Q4: Where was the Design meeting held?							x
Q5: The Movie club has already shown 'Lawrence of Arabia'.	x						
Q6: Which two participants brought ppt presentations to the M Club meeting?				x			
Q7: One of the meetings took place in Geneva.						x	
Q8: Who attended all of the meetings?							x
Q9: Denis proposed a brainstorming area.	x	x	x				
Q10: Who was the marketing expert in the Design meeting?		x					

Table 4.5: Ten questions in the task and the types of search criteria that are relevant for retrieving answers to them

	Content			**Speaker**			**Argumentation**			**Document**		
	V	**MV**	**M**	**V**	**MV**	**M**	**V**	**MV**	**M**	**V**	**MV**	**M**
Q2	1.5	1.13	0.86	1.25	1.1	1.0	0.86	0.38	0.5	-	-	-
Q3	1	0.75	0.38	-	-	-	-	-	-	-	-	-
Q5	1.86	1.36	0.56	-	-	-	-	-	-	-	-	-
Q6	1	0.5	0.38	0.25	0	0	-	-	-	0.75	0.63	0.38
Q9	2	0.86	0.63	0.63	1.0	0.38	0.38	0	0.25	-	-	-
Q10	0.43	0.5	0.67	0.71	1.13	0.5	-	-	-	-	-	-

Table 4.6: Average number of Content, Speaker, Argumentation and Document criteria specified to retrieve answers to questions in the task

Other examples where users chose content search when it was not called for was when questions addressed documents, e.g. Q6 (”Which two participants brought powerpoint presentations to the movie club meeting?”), or speakers, e.g. Q10 (”Who was the marketing expert in the Design meeting?”). In these two examples many users searched for the keywords ”powerpoint” and ”marketing expert” in the meeting discussion, instead of searching for the document type ”slides” or the speaker role ”marketing expert”.

The above results suggest that our two hypotheses are only partially correct. When users express search criteria in natural language, they do tend to select novel criteria more frequently than users who choose search criteria from menus (**H1**). On the other hand, a natural language interface by itself does not incent users to exploit argumentation criteria as often as they are relevant to the search (**H2**). Both with linguistic and non-linguistic interfaces users tend to favour keyword search before other more novel types of search criteria.

4.6.2 Linguistic complexity of queries

The second part of the data analysis concerns the linguistic surface structure of queries. More specifically, what level of linguistic analysis is required in order to correctly interpret queries in the meeting domain (**R2**). The hypothesis that we derived from the work in chapter 3 is that word sense disambiguation, and syntactic role extraction constitute an appropriate input for the domain-specific interpretation rules, which then extract meeting concepts and relations from the question (**H1**). However, the system that we evaluate in this thesis has a graphical user interface attached to it and offers both linguistic and non-linguistic methods for search. Our second hypothesis is that the user interface influences how users express themselves in natural language, namely that their input is syntactically simpler than survey-collected questions, even to the point of expressing single keyword queries (**H2**). This hypothesis is particularly motivated by the flexible multimodal interaction paradigm, described in 4.2.5. The paradigm forces certain simplifications in the language understanding of user input. A natural consequence is that users adapt to those simplifications and express themselves with more simple queries.

For all subjects that had access to a linguistic interface in the first evaluation session (V, VK, MV and MVK), we computed the relative frequency of queries that were expressed as keywords, and queries that had linguistic structure. Notice that ’keyword’ here means that the *surface form* of the query is a single word or a sequence of words without linguistic structure. It does not mean that the query generates a content criterion of type ’keyword’. Queries that are expressed in a keyword style can generate different types of search criteria. For example, the query ”Agnes” generates a speaker criterion, and ”Wednesday” generates a date criterion.

In the first evaluation session we could identify three types of users in our data. The first type preferred keyword style interaction, the second type linguistic interaction, and the third type used both interaction styles in a mixed approach. (see figure 4.15). A user was categorized as preferring a given interaction style

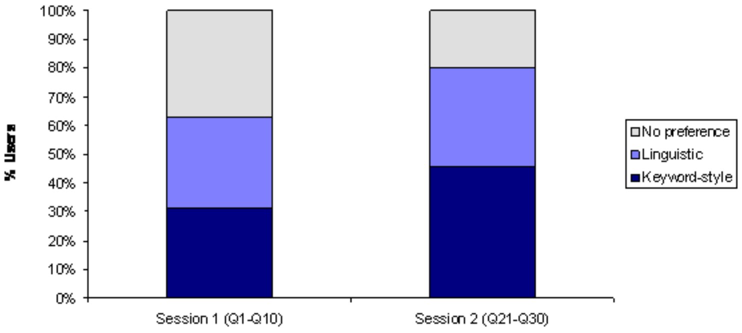


Figure 4.15: User preferences for keyword-style and linguistic querying in the two evaluation sessions

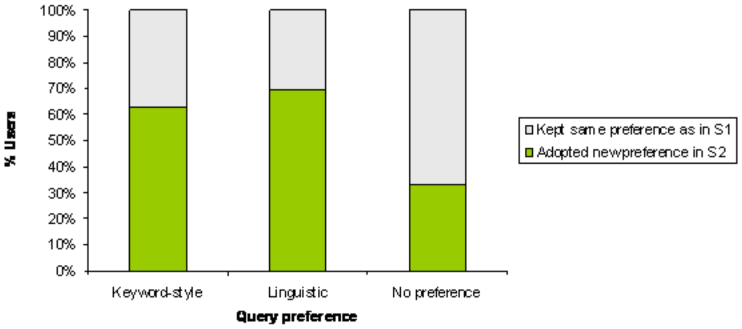


Figure 4.16: Change of preference on query style when going from session 1 to 2

if more than 66% of their input was of that type. The size of the groups was extremely even, which suggests that users' personal preference is a much stronger factor for the choice of interaction style than the design of the user interface.

In the second evaluation session, when users were already familiar with the system, one third of the users kept to the interaction style they had adopted in the first session. The other two thirds converged towards either keyword or linguistic interaction (see figure 4.16). Again, we could not find evidence to confirm our hypothesis that the flexibly multimodal user interface influenced users to express keyword queries rather than linguistic ones (**H2**).

For those users who used a mixed query style in the first session, we tested if there was a correlation between the choice of query style and the question in the task. We used the classification of questions as described in 4.5.3, i.e. we made the distinction between questions where the information to be found was in an

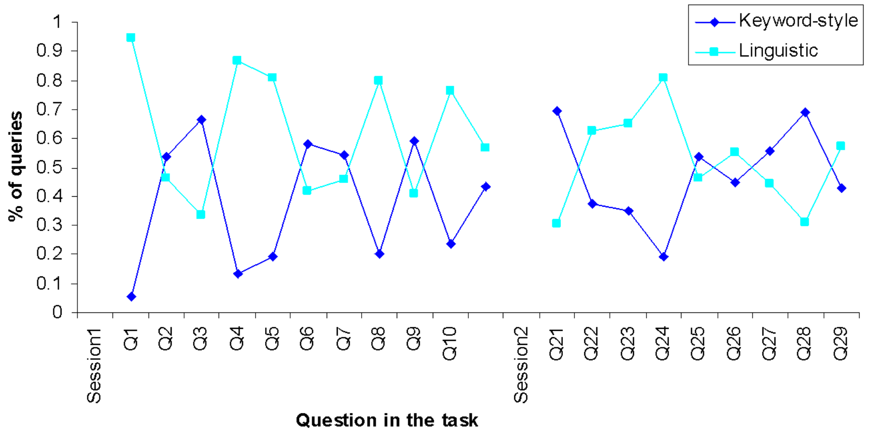


Figure 4.17: Keyword-style and linguistic querying per question in the task (session 1 and 2)

episode of meeting discussion, and questions where the information concerned a speaker, a referenced document, or meta-data such as the date and place of the meeting.

We found that indeed users had two search approaches. In the first session, keyword-style querying was very frequent when the user was searching for content in a meeting transcript, for example in Q3 (”Appliances were discussed in the Furniture 3 meeting”) and Q9 (”Denis proposed a brain-storming area.”) (see figure 4.17). This result is not surprising, considering that most users had extensive experience of document search engines where the input is provided as keywords. The interesting result is that users made linguistic queries when the information they were searching for was not an episode of meeting transcript but rather meta-level information about the meeting, for example Q8 (”Who attended all of the meetings?”) and Q10 (”Who was the marketing expert in the Design meeting?”). For these questions it was foreseen that the user would navigate to that information by accessing the appropriate GUI elements rather than making a query.

In the second session, due to the fact that user preference was very strong for one query style or the other, we could not observe any correlation between query style and question in the task.

The above findings demonstrate two facts. First, they show that the type of natural language input that users provide to a real meeting data retrieval system is more diverse than what was foreseen in chapter 3. Both keyword-style queries and linguistic queries are frequent. Secondly, this diversity is due to both the nature of the information that is being sought for, and users' personal preferences for querying. Hence one cannot conclude that a keyword-based search engine is sufficient for this domain of information retrieval. One can also not design the natural language query interface in such a way that the natural language understanding is strictly based on linguistic analysis of full-sentence questions. Both types of input need to be considered in the design of the NLU component of the system.

4.6.3 Domain-complexity of queries

We have now seen that linguistic queries represent a large fraction of queries to a multimodal meeting data retrieval system. But this alone does not prove that queries need to be analyzed with linguistic resources to obtain a correct domain-specific interpretation of their meaning (**H1** of **R2**). The need for syntactic analysis and syntactic role extraction from user queries is motivated if at least one of the two following properties is true for the query:

1. A meeting concept is disambiguated by its role in a phrase, e.g. a verb phrase.

Query	**Meeting concept**
"Did Denis *show* an [advertising poster]?"	Doctype:poster
"Did they *discuss* the [advertising poster]?"	Topic:poster

2. Several meeting concepts are referred to in the same query, and there is a relation between them.

"[Suggestions] by [Agnes]"	Speaker(agnes)
	Arg_cat(suggestion)
	Make(agnes,suggestion)

To find out how frequent these two phenomena were in queries, we computed the number of search criteria that each linguistic query generated. For those queries that generated only one criterion, we annotated whether there was syntactic information in the phrase that allowed for correctly disambiguating the meeting concept. Notice that during the experiment, a human wizard performed the interpretation manually, using any relevant information that was available, for example what the user saw on the screen and what the user had said previously. Hence, the human interpretations were generally correct, also for queries that did not contain sufficient syntactic information to be disambiguated by linguistic means alone.

The result of our analysis was that we identified four types of queries in the data, summarised in table 4.7. The first type, 27% of the total queries, referred to a single meeting concept which could be disambiguated by using syntactic information in the query. The second type (21%) referred to several meeting concepts and relations in the domain model. The third type (7%) was queries that did not match any meeting concept in the system. Finally, the fourth and most frequent type (45%) was queries that referred to a single meeting concept but could not be disambiguated by any syntactic information in the query. For these queries, and keyword-style queries with no linguistic structure at all, other contextual information is required to disambiguate them.

Reference to meeting concept in linguistic query	**Disambiguated by syntactic role**	**Syntactic roles not applicable for disambiguation**
Single meeting concept	27%	45%
Multiple meeting concepts	21%	0%
No meeting concepts	0%	7%

Table 4.7: Distribution of queries that refer to meeting concepts in the elicited query set

In view of the fact that so few queries referred to more than one meeting concept, one may argue that the natural language understanding of queries does not require syntactic role extraction, or subsequent extraction of domain-specific relations between meeting concepts (**H1** of **R2**). However, we need to consider the experimental method used in the evaluation, in particular how the questions were designed for the task. Each question is very specific and imposes a limit on the possible search criteria that users can specify to find the answer. 7 of the 19 questions solved during the evaluation were formulated in such a way that only one search criterion was required to find the answer. 45% of all the single-concept queries were expressed while solving those 7 questions. A more appropriate approach for eliciting user queries for the design of syntactic and semantic natural language understanding components would be to define a vaguer task for the experiment. In that task, users would have to decide for themselves what pieces of information to search for in the meeting discussions. The problem is that such tasks are very difficult to create for laboratory experiments, because subjects in the experiment are not familiar with the meeting discussions in the database and do not have the proper intrinsic motivation to solve the task. Field studies would be interesting in this case.

An important result is that although a majority of queries refer to only one meeting concept, they are not trivial to disambiguate. Queries are made in a context, both in terms of previous interaction and the information that the user sees on the screen. Interesting future research would be to take this contextual information into account when developing natural language understanding engines for queries in this domain.

4.6.4 Sequential queries

In the previous section we found that a surprisingly low number of queries refer to more than one meeting concept. In 4.6.1 we also learned that keyword-style queries are very frequent. Hence, full-sentence questions that generate several search criteria in one interaction are only a small subset of all the natural language queries that users make to a multimodal meeting data retrieval system. One possible reason for the single-concept queries is that the system allows sequential interaction, i.e. refinement of search by expressing additional queries. Such a search approach was not considered at all in the design of the natural language query engine in chapter 3. Sequential search can be an advantage for users who do not know exactly what to search for from the start. When they

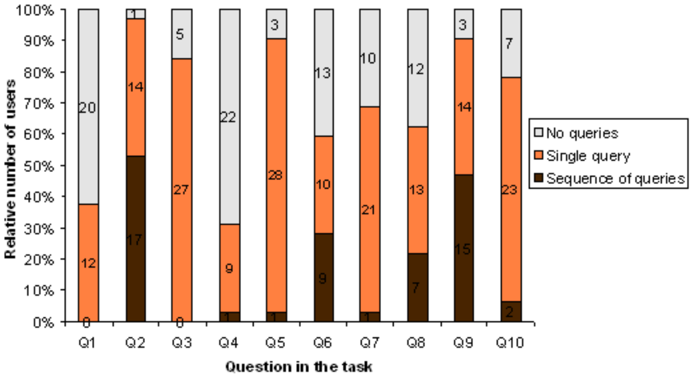


Figure 4.18: Relative number of users who search with multiple queries, a single query and who only browse

have obtained results (maybe browsed through some), they may discover which additional queries they need to make to find the desired data. In this section we analyze how frequent such sequential interactions are.

For each question in the task, we grouped users according to their search approach. The three search approaches considered were:

1. The user performed the search without expressing any queries in natural language
2. The user specified all desired search criteria in one query
3. The user searched with two or more sequential queries

The results are summarized in figure 4.18. We found that questions Q2, Q6, Q8 and Q9 were of the kind where all three search approaches were used. For the other question only approach 1 and 2 were used, which can be explained by the nature of those questions. Either navigation in the graphical user interface was sufficient to find the answer, or only one search criterion was needed to find the relevant piece of meeting discussion. For those four questions where sequential querying occurred, between 25-55% of users chose that approach, depending on the question in the task. But there was also a large second group of users who chose to search with a single, more complex query. The important message here is that when a task consists in specifying multiple search criteria, sequential querying is as frequent as full-sentence linguistic querying. And when users make queries sequentially the queries are less complex, both in terms of linguistic structure and in terms of reference to concepts and relations in the meeting domain.

An additional explanation for the high amount of single-concept queries is repetitive search behaviour. Not all queries, made one after the other, count as proper query sequences i.e. progressive refinement of the search. We found many cases where the user started with a full-sentence linguistic query that generated multiple search criteria, and then made a shorter query, generating a subset of the same search criteria. These repetitions were quite frequent (15-25% of queries) and, from what we could observe, they were caused by many unrelated reasons, for example that the first query did not generate the expected search results, that the system (wizard) did not hear the query properly, or that a desired search criterion did not exist in the system.

4.6.5 Natural language querying vs. menu-selection

In this section we enter into the second objective of the Wizard of Oz evaluation, namely to identify properties in natural language that motivate the integration of this modality into meeting data retrieval interfaces (**R3**). The two hypotheses that we test are: whether natural language querying is more efficient for search than tactile menu-selection when the user is not sure which precise attributes and values that best reflect what they are searching for (**H1**) and whether natural language is more efficient when the range of possible values is unrestricted, and the user knows which value they want to select for their task (**H2**).

To test these hypotheses, we compare three modality conditions: voice-only (V), mouse-only (M), and combined mouse-voice (MV). In the first condition, users are forced to make all their interaction with the system in natural language, but they can select criteria by querying freely, e.g. "show meetings where appliances were discussed", or by performing series of GUI actions with voice-commands to select the search criteria from menus, e.g. "Open content criteria", "Open topic", "appliances". In the second condition (M), users are forced to select from menus. In the third (MV), users are free to choose.

The first difference we found between the completely menu-based condition (M) and the natural language-based conditions (V and MV) was that users in the M condition more often solved questions without specifying any search criteria at all. Instead they accessed the data directly and browsed until they found the answer (see figure 4.19). This was particularly evident for Q3: "*Appliances were discussed in the Furniture 3 meeting*", but also for other questions it was relatively frequent. In the total task (Q1-Q10), 40% of the questions were solved without search criteria in condition M, whereas in the linguistic conditions V and MV, the ration was 20% and 22% respectively. The browsing-only behaviour can have at least two reasons: either that menu-selection of criteria required so much effort that users found it more convenient to browse the data directly, or that users did not know which search criteria button to choose, so they chose none. In both cases it appears as if a natural language query interface provides a useful alternative to menus, as it incents users to specified search criteria before browsing the data. Our hypothesis that natural language is more efficient than menus when users are not sure of what criteria to specify (**H1**) is supported by the above results.

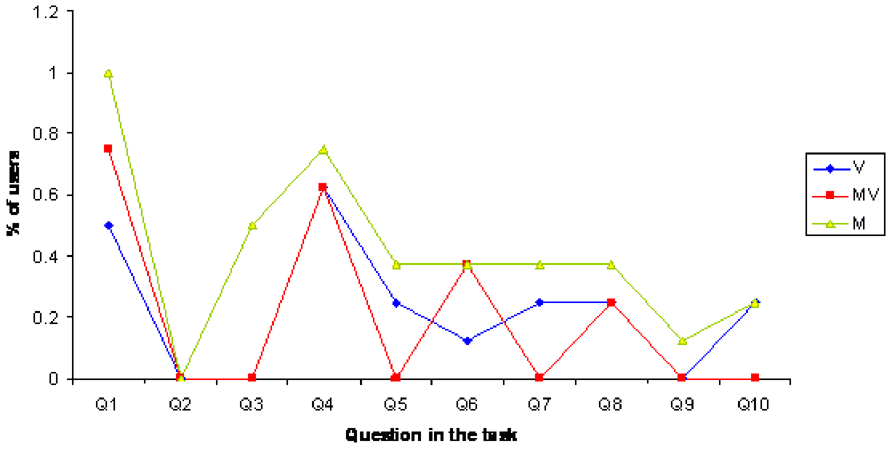


Figure 4.19: Percentage of users who solved questions in the task without specifying search criteria

Next we compared the V and MV conditions to see whether users chose to select criteria by querying or by menu-selection. The results show that users who made all their interactions with the system in natural language (V) mainly chose to make queries, whereas users who had access to both natural language and tactile manipulation (MV) more often chose menus (see figures 4.20 and 4.21). This was particularly evident for topic and keyword criteria. For the other criteria V users consistently chose querying, whereas MV users chose both querying and menu-selection.

We would have expected both groups to favour querying above menus, because querying requires fewer interactions with the system (no need to first access and browse menus). We would in particular have expected users to favour querying when selecting topic and keyword criteria. These two represent the type of attributes that are hard to distinguish one from the other and hence make users uncertain about which one to choose for their search (H1). They are also representative of the type of attributes for which the ranges of possible values are unrestricted (H2). In other words, our results do not show any correlation between the choice to make NL queries and the types of attribute-value pairs to be specified. H1 and H2 are not supported. In fact, from observing the recordings of the evaluations, we suspect that the choice between querying and menu-selection does not have to do with efficiency at all. It is rather the case that menus are considered as the ”safer” option since the user is able to see the possible values for each attribute and is in full control of which attribute-value pairs are generated. When querying, the system automatically generates the attribute-value pairs from the language input, and during this interpretation process many things can go wrong, for example: 1) that the system (i.e. wizard) does not hear what the user says, 2) that the requested criteria are not available in the system or 3) that the generated criteria are not the ones the user intended.

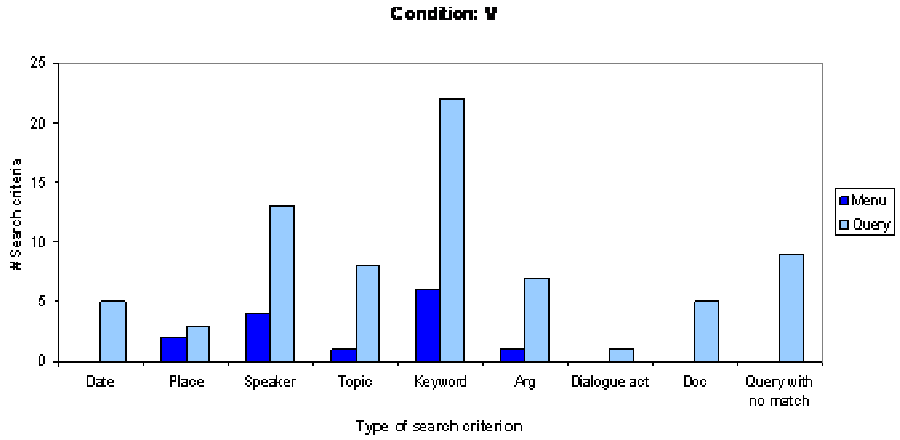


Figure 4.20: Search criteria generated with menu-selection vs. free natural language querying in the V condition

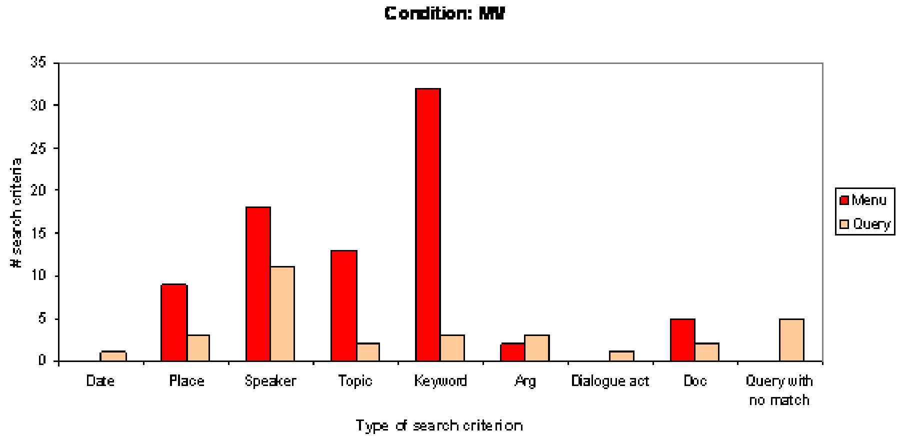


Figure 4.21: Search criteria generated with menu-selection vs. free natural language querying in the MV condition

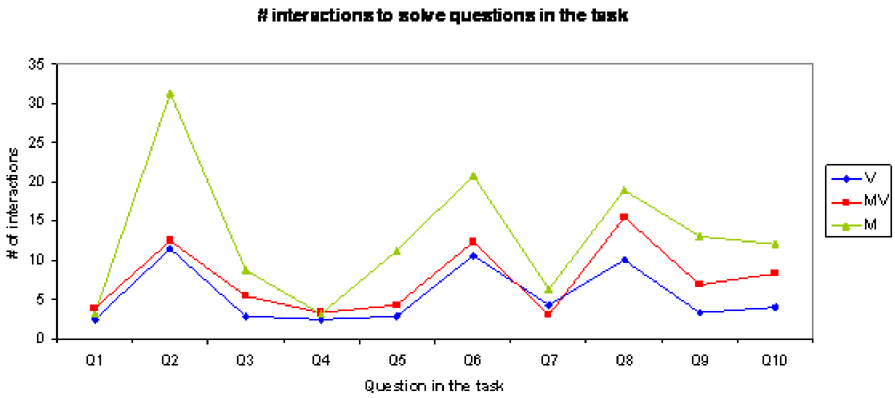


Figure 4.22: Number of interaction steps taken to solve the task

Such uncertainties in natural language processing might be seen as a bigger disadvantage than the effort of accessing, browsing, and selecting values from menus, hence incenting users to choose the approach that provides the most predictable system behaviour but not necessarily the most efficient way of using of the system.

To confirm that MV users indeed need more interaction steps than V users to solve the task, we counted the exact number of interactions that users made in each question. The results can be seen in figure 4.22. Users in MV made consistently but only marginally more interactions than users in V (between 1-5 interactions more per question). A likely reason for this small difference is that the counting includes interactions for both search and browsing of results. The book-browsing actions are equal for all modality conditions, and they represent a very large fraction of the total interactions.

We also compared the results with the M condition and found that users that had access to both natural language and tactile manipulation (MV) made considerably fewer interactions than users who could only interact with tactile manipulation (M). This was unexpected since users in MV favoured menu-based criteria selection and hence should have solved the question with approximately the same number of interactions as users in M. Some possible explanations are that users in M more often accessed menus and browsed them without selecting any value, or that users in the MV condition accessed a menu and then used a voice-command to select a value far down the list not visible on the screen, hence skipping the list-browsing step. In any case, the result in figure 4.22 suggests that natural language does contribute to efficiency in terms of number of interactions, even though we have not identified what those precise short-cuts are.

Elaborating on the (in)efficiency of menu-based selection of criteria, it is striking that in the V condition menus were almost entirely ignored by users. In this

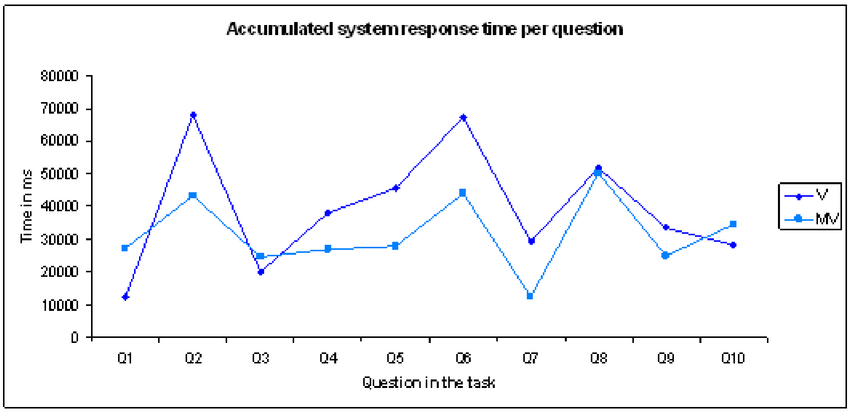


Figure 4.23: System response time in the V and MV conditions

condition querying must have been extremely more efficient than menus, despite all the potential NLP errors that could occur through free NL querying. We believe that the preference was related to system response time. In order to select from menus, users would have to make voice commands to access the desired menu, then to scroll to the relevant value, and finally to say the value. Performing GUI actions by voice is notably less efficient than by mouse. The system has to process the language input each time, which causes delay in response time. It appears that this delay weighed heavier as a disadvantage than the potential errors in the NLP of queries, hence incenting users to choose the more error-prone approach while saving interaction time. Figure 4.23 shows the accumulated system response time for interactions during questions in the task. For most questions the system response time for V was higher than for MV. It is an indicator of the fact that users in V had more motivation to choose search approaches that minimized the accumulated system response time. Interestingly, in the post-evaluation questionnaire users in the V condition reported a higher satisfaction with the system's reaction time than users in the other conditions. They also reported that they did not find the search criteria buttons very useful, whereas users in other conditions reported that they did.

Finally, looking at the overall task performance (see table 4.8), we can see that users in the V condition solved more questions, and more often got a high score on their answers to questions than in the other two conditions (M and MV). Since this was the only group that consistently searched by querying, it seems that this search approach is more efficient than menu-selection and leads to more accurate retrieval of answers to questions from meeting data.

Summing up the results of this section, we have been able to show that users who interact with a GUI-based meeting data retrieval system in natural language (V) prefer to search for information by expressing natural language queries, whereas users who can choose between natural language and tactile interaction often choose menus. In other words we have not been able to show that natural language has a property that universally incents users to favour natu-

	V	**MV**	**M**
Average number of questions solved	10.9	9.8	10.1
Average score	2.6	2.6	2.4
Score 0 (%)	6.9	8.6	12.8
Score 1 (%)	4.6	2.5	7.7
Score 2 (%)	8.0	12.3	2.6
Score 3 (%)	80.0	76.5	76.9

Table 4.8: Task performance

ral language querying over menu-selection in this domain. However, we still argue that linguistic query interfaces are useful in multimodal meeting data retrieval systems. In terms of efficiency, we have seen that the integration of natural language leads to fewer interaction steps to solve the task. Users are also more accurate in retrieving answers to questions. The problem with the tested Archivus user interface is in our opinion that it does not sufficiently support NL interaction. The interpretation of queries happens in a "black box". The system gives only minimal feedback on how queries are interpreted, namely by displaying search criteria in the Current search criteria list. It can easily happen that the generated criteria represent only part of what the user requested, either because there was no match in the system database for some specific term, or because the input was not properly recognized by the ASR. A typical example that occurred in the evaluation was that users made the query "Which meetings happened on March tenth?". The search criterion "Month:March" appeared in the Current search criteria list, but "DayOf Month:10" did not, because none of the meetings in the database was held at that precise date. In the post-evaluation questionnaire, when users were asked to describe in what way the Current search criteria was most useful to them, only 3% answered that it showed how the natural language query had been interpreted. The other answers related to search strategies, e.g. that the Current search criteria gave a reminder of which criteria had already been specified (36%), and that the list was useful for refining the search (29%) and for deleting unnecessary criteria (21%).

Another aspect that incents users to access menus instead of querying the system freely is that menus are the only source of information that provides a clear scope of the attributes and values that the system can produce as a result of an interpretation. Even if a user prefers querying, they may access the menus to learn the scope of the system. An interesting alternative to the current design of the Archivus interface is to dedicate more work into the Current search criteria component of the system, i.e. make it more informative and interactive, so that users have a possibility to review in more detail how their queries are

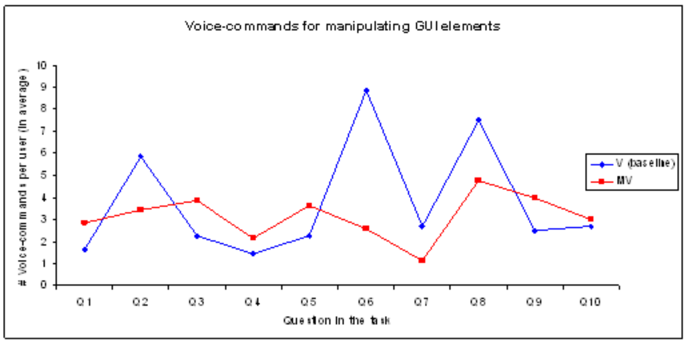


Figure 4.24: The average use of voice-commands for manipulating graphical components of the user interface

interpreted. In chapter 5 we address this issue by comparing two versions of the system: one in which users have a more detailed Current search criteria list and are only able to search with natural language, and one in which users have access to menus but are only able to search with a tactile modality.

4.6.6 Manipulations of the GUI with voice-commands

So far the results in this chapter have focused on natural language querying and generation of search criteria. Another property of natural language that is considered in the research questions is the ability to refer to GUI elements that are not visible on the screen. As Archivus is flexibly multimodal, users can manipulate all GUI elements such as buttons and arrows with voice-commands instead of mouse. Search criteria menus are one example, but also the search results can be browsed with voice commands, e.g. by opening books ("Open the Furniture 1 meeting"), accessing sections of the book with content tabs ("Show me the appendix"), and browsing pages ("next page", previous page"). In this section we address the hypothesis about whether natural language makes GUI navigation more efficient by allowing short-cuts to GUI elements that are not visible on the screen (**H3**). For standard GUI manipulations such as selecting buttons that are visible on the screen, we assume that mouse is more efficient.

As a first test, we compare the two conditions V and MV to see how often users in the MV condition choose to make voice-commands, as opposed to users in V who have to do them in order to access search results. Our results show that for 6 questions out of 10 in the task, users in MV actually make more voice-commands than users in V (see figure 4.24). However, only 4% of the voice-commands are short-cuts to GUI elements not displayed on the screen. In the remaining 96% of cases, users simply prefer to make the GUI manipulation by voice instead of mouse. The result contradicts our hypothesis (H3) by showing that users make voice-commands even when voice-commands do not contribute to efficiency in any apparent way.

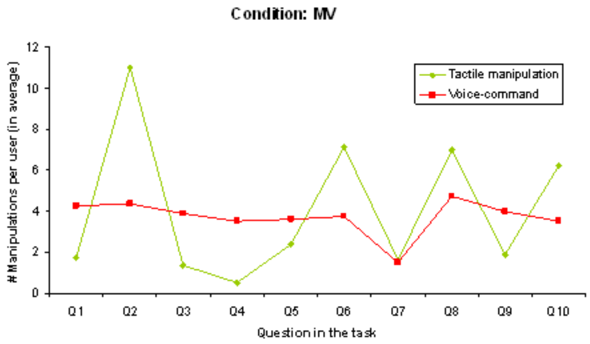


Figure 4.25: Use of mouse and voice for manipulating elements of the GUI

To get a clearer view of how often users actually choose voice-commands over tactile manipulations, we count the distribution for each question in the task. In the result (see figure 4.25) we can see that for half of the questions users make more voice-commands than tactile manipulations. During the whole task the distribution is 52% mouse and 48% voice.

The result signifies that natural language is not used only for efficiency, but also due to other factors that we cannot find in this data, for example that it gives pleasure. In the previous section we saw the same tendency. Users who had access to both tactile and language modalities did not use language for querying, which was foreseen as the most efficient way of selecting search criteria, but instead used search criteria menus.

Our final hypothesis on which properties motivate the integration of natural language in multimodal meeting retrieval systems is that speech is useful for controlling the GUI when the user's hands are busy, for example making notes with a pen (**H4**). In the evaluation task, the most typical situation in which users' hands were busy was when they wrote down the answer to a question. Once they had done that, they were instructed to reset the system with a special "Task finished" button to clear the search results before moving on to the next question. We observed that this was a situation in which users interacted with the system while being busy with their hands. In 90% of the cases when users restarted the system, they said "Task finished" as a voice-command, whereas the "Task-finished" button was selected with mouse only in 10% of cases.

4.6.7 General satisfaction with natural language interaction

In this last section we address the hypothesis that relates to user satisfaction, namely whether natural language contributes to higher satisfaction with the general functionality of a meeting data retrieval system, even when it does not lead to more efficient retrieval (**H5**). By general satisfaction we consider aspects like feeling in control of the system, finding the system easy to use etc.

We have already seen that natural language is frequently used in situations where it does not lead to more efficient retrieval. The question that follows is: why did users so often choose natural language over tactile manipulation, if it was not more efficient? To evaluate user satisfaction, we compare responses given in the post-evaluation questionnaire. It is important to note that the questionnaire was answered after the user had undergone two sessions with the system, first in one of the modality conditions (V, MV, etc.) and then with the full set of modalities (MVK). This means that also users in the M condition had access to speech during the second half of the evaluation, so the answers in the questionnaires cannot be compared strictly by condition, e.g. that users with a natural language modality were more satisfied with the system than users who had access to mouse only. On the other hand, we believe that the general opinion of the system strongly stabilizes already during the first evaluation session, when users learn to use the system with a specific subset of modalities. Hence, in the later phase when more modalities are added to the system, this general opinion is not likely to change dramatically, and the responses in the questionnaire can be considered to represent the first evaluation session.

In terms of general system usability, the results of the questionnaire were the following: users who started using the system with mouse (M) found it easier to *learn* the system functionalities than users who also had to learn how to interact in natural language (V, MV and VK). On the other hand, users who had access to natural language from the start found it easier to *use* the system than users who had access only to tactile manipulation. Furthermore, users who started using the system with only natural language (V, VK) felt more in *control* of the system than users who used it with tactile manipulation (M, MV).

When asked about the usefulness of the two modalities mouse and voice, we could observe a certain influence from the modality conditions in the first evaluation session. All users, regardless of condition found that voice control was useful. However, when asked to rank the modalities by usefulness, those users who had learnt the system with only natural language (V, VK) found the voice-modality more useful than users who had access to mouse (M, MV). Voice was declared as useful both for finding information in books and for browsing books. Contrary, those users who had learnt the system with only mouse, ranked this modality higher, both for finding information in books and browsing them.

To further evaluate the importance that users gave to the natural language modality, we categorized answers to the open-ended question ”What did you like most about the system?”. Surprisingly many of the answers referred to the speech interface (34%). Other typical answers related to the richness, organization and annotation of the meeting data, which were also novelties in the domain of information retrieval (see table 4.9).

To find out if any of the negative impressions of the system were related to the natural language interface, we also categorized answers to the open-ended question ”What did you like least about the system?”. The most frequent answer referred to the system’s response time (34%), but not exclusively to the interpretation of natural language requests. Also tactile manipulations such as

Ratio of answers	**Aspect of the system**	**Motivation given by experiment subjects**
34%	Speech interface	You could ask questions, control the user interface with voice, and the speech recognition performance was very good.
24%	Organization of media	You could browse the transcript and video in parallel, and the documents were linked to the meeting transcript.
20%	Annotations on data	You could find specific episodes based on topic search or other criteria, and the highlighting on pages helped to find the answer
12%	Richness of data	The meetings were transcribed, and also the recordings and documents were available in the system.
10%	Flexible multimodality	You could choose different approaches for performing the same task

Table 4.9: Five typical answers given to the open-ended question ”What did you like most about the system?”

Ratio of answers	**Aspect of the system**	**Motivation given by experiment subjects**
34%	System response time	The system reacted slowly to requests, and it was slow in loading meeting books.
17%	System control	It was not easy or possible to go back to the previous system state, e.g. to jump back to a previously visited page of the book after leaving the book
15%	Search criteria menus	They were difficult to use. It was hard to choose the relevant criteria and the hierarchical access to the values was complicated (e.g. first Content, then Topic or Keyword).
10%	Browsing meeting books	The tabs and arrows were not easy to use, or sufficient tools for browsing.

Table 4.10: Four typical answers given to the open-ended question "What did you like least about the system?"

opening meeting books caused delay in response time. The other answers concerned the design and functionality of the system in general, not the interaction modalities (see table 4.10).

These results show that users who had the opportunity to use natural language from the very beginning of the experiment were very satisfied with this modality, and also with the system in general. Any negative experiences, such as frustrations over speech recognition failures, were amply compensated by the positive experience of being able to interact in natural language. One additional indicator that natural language contributed to higher system satisfaction (**H5**) is that users who had learnt to use the system with natural language were prepared to use the system again in the future, whereas users who had learnt it with mouse were less eager. So, from a usability point of view, including natural language as an interaction modality in meeting data retrieval systems is motivated by the fact that using it leads to at least one of the following: higher task performance, fewer interactions or more enthusiasm in using the system (**R2**).

4.7 Conclusions of the chapter

The objective of this chapter was to evaluate users' natural language interaction with a multimodal system for meeting data retrieval in order to gain insights about how natural language can be beneficial in this domain of information retrieval.

Previous survey studies have shown that one of the strengths of natural language is the possibility to ask complex questions about the higher-level aspects of discussions in a very natural and precise way; and that questions about the argumentative process are indeed what users want to ask. In our evaluation, we observed the opposite behaviour. Users were in general cautious about asking complex questions to the system, partially because they did not know the limitations of the natural language understanding. Also, users did not refer to the argumentative aspects of the discussion as much as we expected but instead tried to work heavily with topic and keyword search, as this is the type of search most users are familiar with. We conclude that it takes more than the availability of argumentative annotation of discussions and a natural language interface to that data to make people ask more complex questions about discussions and to become efficient in retrieving information from spontaneous, spoken conversations. As an extension of the work, we believe that providing users an informative overview of the discussion and the annotations made on it can provide incentive to exploit the argumentative annotations more during search.

The natural language understanding of queries in this domain is, in theory, dependent on deep linguistic analysis. If users ask the type of questions that were collected in survey studies, the natural language processing should involve at least word sense disambiguation against a domain-specific semantic lexicon, and syntactic analysis with assignment of syntactic roles. In practice however, we have seen that queries to a multimodal graphical user interface for meeting data retrieval are often short and have very simple linguistic structure, if any at all. Also, they tend to be sequential. Therefore, we conclude that syntactic analysis is applicable for interpreting only a subset of queries in this type of meeting data retrieval system. For the majority of queries, the natural language understanding should take into account contextual information, in particular the history of previous queries and the navigational actions in the graphical user interface. How to integrate this contextual information into the interpretation rules of the natural language understanding engine is a target for further study.

As a competitive modality to tactile manipulation, natural language is not used exclusively to make the search more efficient. For example users do not consistently choose free querying over menu-selection. Voice-commands are also used for manipulating GUI components. From the results we obtained, we conclude that natural language brings several advantages in the context of meeting data retrieval: it incents users to express more search criteria; reduces interaction steps to solve the task; increases the accuracy in retrieving meeting data; and contributes to general system satisfaction.

5 Visualizing meeting discussions

Searching for information in meeting discussions with natural language queries is not as intuitive at it may seem. Factors such as the graphical design of the user interface, and the limited scope of the system's natural language understanding, incent users to express simple queries, and mainly content-oriented ones. Although meeting discussions have been annotated with multiple layers of higher-level structures to make the information retrieval more efficient, users do not exploit these annotations when expressing queries. In this chapter we propose to enhance the task with visual meeting overviews that display the annotated higher-level structures of meetings. More specifically, we compare two types of meeting overviews, topic overviews and conversation graphs, and show that the compact representation of topics, speakers and argumentation in conversation graphs enhances both querying and browsing of meeting data. Users exploit the annotations in a targeted manner when expressing queries to the system, do not repeat queries as frequently as when working with a topic overview, and more often retrieve the relevant episodes of meeting discussions to answer questions in the task. We conclude that conversation graphs are helpful aids for searching in conversational data, and propose as future work to integrate them as interactive components of meeting data retrieval systems.

5.1 Introduction

The work described in this chapter is motivated by the outcomes of the previous two chapters. The overall goal has been to understand how natural language can be used for searching information in recorded spontaneous conversations. In chapter 3 we outlined the architecture of a natural language query engine to meeting data, addressing the issues of annotating meeting discussions to make the search more efficient, and developing linguistic and domain-specific techniques for understanding questions on such annotated discussions. In chapter 4 we evaluated a multimodal meeting retrieval and browsing system, Archivus, with the Wizard of Oz method in order to learn whether users asked the type of complex questions with linguistic structure that had been foreseen in the design of the query engine in chapter 3, whether they favoured natural language search over menu-based search; and whether the available annotations in the meeting data were exploited.

Although natural language was shown to be a highly appropriate and appreciated modality for searching in conversational data, there were many situations in which it was not used to its full potential. One such situation was the specification of multiple search criteria. Very few users chose to specify several criteria in a single query, and instead went through the trouble to access various menus, or request one search criterion at a time. Another situation where natural language was not used as had been foreseen, was when searching for argumentative segments in meeting discussions. The argumentative annotation layer was not sufficiently exploited when expressing queries to the system.

A possible reason for the suboptimal use of natural language in this domain of information retrieval is that the design of the user interface and the presentation of the meeting data did not provide sufficient support for guiding users in their formulation of queries. Also, there were several novelties in the system that made the task more difficult. Users were not previously familiar with higher-level annotations on conversational data, and most users had not used natural language as a modality for search tasks. Moreover, the flexible multimodality gave users different choices of interaction at every step of the task. Consequently, it was not instantly obvious how to use natural language in this application.

An important biasing factor was that the meeting discussions were presented as text documents. Only the 'raw' data (transcripts) was displayed when opening a meeting. The higher-level annotations were by default hidden. Only when users searched for the annotations, they became highlighted in the transcript, but even then only the segmentation, not the labels. This type of text-oriented visualization of the data most likely influences users to search with standard information retrieval techniques, i.e. to specify keywords and topics but not argumentative categories, dialogue acts and speakers.

In this chapter the goal is to propose solutions to the above problems. We have two concrete objectives. The first is to test different visualizations of meeting data to evaluate their impact on the natural language meeting data retrieval task. More specifically, we want to compare two types of overviews: a table of contents-style topic overview and a new form of overview called conversation graphs, and to test if the form and content of the visual meeting overview influences to what extent users exploit the higher-level annotations in the data and express complex queries to the system. With complex queries we here refer to queries that instantiate multiple concepts and relations in the meeting domain.

The second objective is to compare natural language and menu-based search in more strict and controlled conditions, to evaluate if the two modalities are used in a more efficient manner when the flexible multimodality is removed and users have only one technique available. To this end, we design two new versions of the Archivus prototype, one with a natural language query interface, and one with a menu-based interface. In both interfaces, mouse is available for browsing the data. It is hence still a multimodal system, but not a flexible one. In the natural-language enabled version, the use of the language modality is strictly limited to querying, whereas mouse is provided for browsing. In the second version, the system is unimodal, offering only mouse interaction. The menu-based version of the system serves as a baseline condition, to determine whether meeting overviews or search techniques have the more dominant influence on users' search behaviour.

The rest of the chapter is structured as follows: first we motivate the need for meeting overviews in the meeting data retrieval task in general, and describe the two meeting overviews used in this work (5.2). Then we outline the concrete research questions and hypotheses that we want to validate by performing Wizard of Oz evaluations with the two meeting overviews (5.3). The experimental method used for the evaluation is similar to the one used in chapter 4, but with new user interfaces and evaluation conditions, and some modifications to the task and meeting data (5.4). In the experimental results we show that conversation graphs are used more consistently than topic overviews both as support for querying and for browsing search results, and that users who work with conversation graphs get a higher task performance. We also validate that the natural language and menu-based search techniques are equivalent in terms of what search criteria users choose and how successful they are in solving the task (5.5). We conclude that conversation graphs are helpful aids for searching in conversational data, and propose as future work to integrate them as interactive components of meeting data retrieval systems (5.7).

5.2 Visualizing the annotation layers of meeting discussions

5.2.1 Motivation

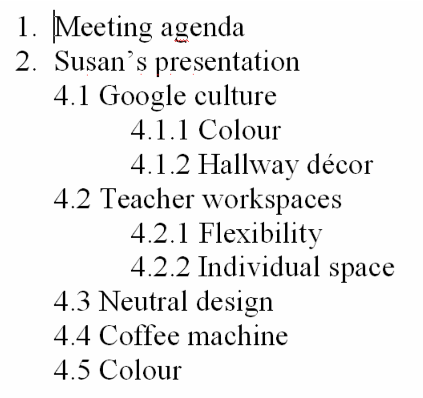
As we have shown in chapter 3, meeting databases can be very complex and contain multiple data layers, created by expert annotators who have post-processed meeting recordings and added higher-level structure to meeting discussions. In order for users to take advantage of these different data-processing outputs and annotations, they need to be aware of their existence when searching in this data. For example, if a meeting recording has been transcribed, segmented into topical episodes and indexed with argumentative contributions made by the speakers, the user can search for episodes where a given person proposed an idea or where a decision was made about a given issue. To make users aware of these annotation layers, meeting overviews can be made available that show what happened in the discussion. The idea is that users can use such meeting overviews as a reference when formulating queries or when browsing the recording or transcript. From a cognitive point of view, meeting overviews can be seen as a kind of 'cognitive artefacts' - artificial devices designed to display information in order to serve a representational function. It has been shown that cognitive artefacts in general enhance the task a user has to do, without enhancing the cognitive load of the task (Norman, 1991; Hutchins, 1995). We hence have a grounded cognitive motivation for introducing meeting overviews into the meeting data retrieval task.

How to create a meeting overview, and what information to include in it, depends on the available annotations on the data, but also on the readability and usability of the overview. If there is too much or too little detail in the overview, the user may not be able to understand it or extract relevant information from it. In this chapter the data we consider is meeting transcripts that have been segmented into topics, and where utterances produced by the participants have been annotated with the identification of the speaker and their argumentative contribution. We compare two types of meeting overviews, a simple text-based topic overview, and a graphical overview showing topics, speakers and argumentation. The goal is to find out if the form and content of the meeting overview has an impact on how users search in meeting discussions. This is particularly relevant in the assessment of the need of adding higher-level annotations to meeting data. Discourse annotations represent a considerable effort in the development of meeting data retrieval systems. If our work shows that a certain type of meeting overview triggers users to exploit the higher-level annotations more, and that the exploitation of these annotations makes them more efficient in performing their task, we have made a contribution to the debate about whether or not it is worthwhile to invest effort in making such annotations.

5.2.2 Topic overviews

The first type of meeting overview that we want to evaluate is topic overviews. Topic overviews are simple, text-based "summaries" of meetings that are generated from the flat or hierarchical topical segmentation of a meeting (see figure 5.1). They are a kind of table of contents of the meeting, similar to a meeting agenda. The difference is that topic overviews are generated from the actual meeting discussion, which means that the same topic can occur several times or that new topics, not present in the agenda, can appear, all depending on what the participants said during the meeting. 'Table of contents' is a standard way of summarizing textual documents, and in the case of conversational data such as meeting discussions, many research projects have been involved in segmenting and labeling the topics of discussion (Galley and Mckeown, 2003; Gruenstein *et al.*, 2005; Banerjee and Rudnicky, 2007). The topical indexing has also been integrated as an interactive component of meeting browsers (Ailomaa *et al.*, 2006; Michaelides *et al.*, 2006; Popescu-Belis and Gorgescul, 2006). The aim of these interactive topical overviews is to make the browsing more efficient by providing short-cuts to the episodes that are topically relevant to a given search task.

In this work our main interest in topic overviews is for search rather than browsing. We want to find out whether topic overviews are useful aids for formulating queries to a natural language-based meeting data retrieval system. If a user for example wants to find out if there was agreement about the choice of colour scheme in a meeting about furnishing a room, a topic overview can be useful by showing that "colour" was discussed several times during the meeting. One straight-forward way to use this information is to ask the system "Show me the discussions about colour". If the meeting is also annotated with argumentative contributions such as agreements and disagreements, the user has the possibility to be more specific and search directly for "agreements about colour",



1. Meeting agenda
2. Susan's presentation
    4.1 Google culture
        4.1.1 Colour
        4.1.2 Hallway décor
    4.2 Teacher workspaces
        4.2.1 Flexibility
        4.2.2 Individual space
    4.3 Neutral design
    4.4 Coffee machine
    4.5 Colour

Figure 5.1: Piece of a topic overview from a meeting about room furnishing

but since topic overviews only show one type of annotation - the topical segmentation of the meeting - we believe that the user will not take the additional, not visualized, annotations into account but only search for "colour" and then read the transcript or watch the recording to determine where agreements and disagreements took place.

5.2.3 Conversation graphs

To visualize several different kinds of annotations made on meeting data we propose a new form of meeting overview, hereafter referred to as conversation graphs. The idea of visualizing the structure of conversations in graph form is not a new one. One of the early examples is the structuring of ancient Greek storylines (see figure 5.2). Although Greek story lines are not exactly human-human dialogues but also include the narrator and development of events, it has some interesting analogies with the structure of meeting discussions: there are a set of characters who share a situation (they are gathered together to make a decision about something) and who share a surrounding (the meeting room). There is a moment in which a conflict arises (someone rejects the proposal of another meeting participant) and a possibly heated discussion follows. The climax is reached when a decision gets made. Then practical aspects of the decision are discussed, such as task assignments. Here more disagreements can occur. The meeting ends when all issues have been resolved and a date for the next meeting has been settled.

Of course, in real meetings the structure is not as static as in the above description. There can be many decisions about different issues, and some issues may not be resolved at all. But the graphical representation provides an informative visualization of what happened in the meeting. It lifts out the argumentative aspect of the meeting content, and makes it concrete and searchable. We take inspiration of this graph representation in our design of conversation graphs for meeting discussions.

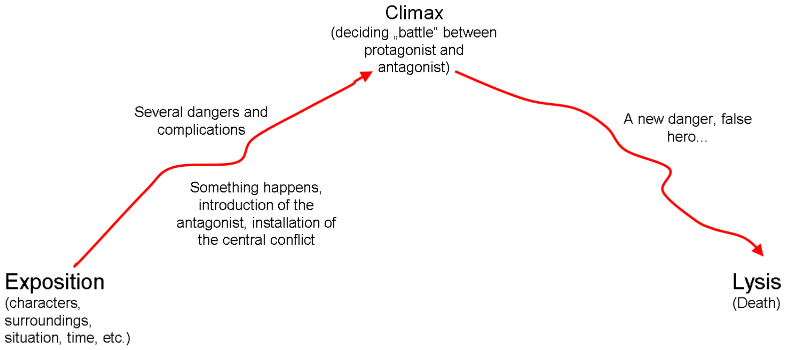

Climax
(deciding „battle“ between protagonist and antagonist)
Several dangers and complications
A new danger, false hero...
Something happens, introduction of the antagonist, installation of the central conflict
Exposition
(characters, surroundings, situation, time, etc.)
Lysis
(Death)

Figure 5.2: Graph representation of the structure of ancient Greek story lines

The conversation graphs that we propose are diagrams that summarize the different annotation layers that are available on recorded meeting discussions, namely what topics were discussed, but also how long they were discussed, which participants were involved in the discussion, and what type of arguments they contributed with (see figure 5.3).

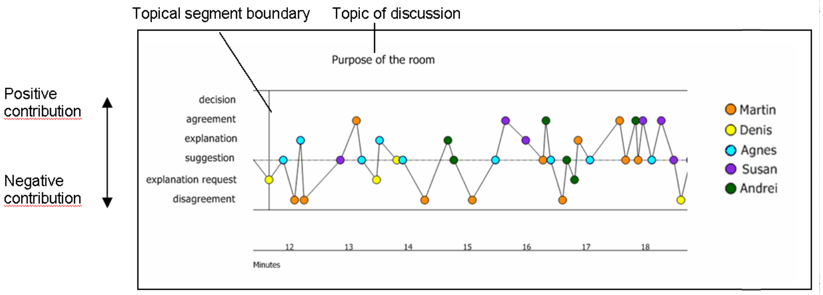


Figure 5.3: Piece of a conversation graph showing a discussion about the purpose of a room

As we have previously mentioned, argumentative annotation of meeting discussions is relatively novel in the field of meeting data retrieval and browsing, and there are no standard annotation schemas for structuring meeting discussions, or for visualizing the structure (Pallotta *et al.*, 2004; Verbree, 2006). The choice of annotation schema, as well as the choice of visualization, depends on what the annotations are intended for. Previous work on visualizing argumentation has mainly been driven by the need for tools to improve argumentation in real-time meetings (Bachler *et al.*, 2003; Fujita *et al.*, 1998; Michaelides *et al.*, 2006; Rienks *et al.*, 2005). Some research has also addressed the use of such visualizations for browsing past meetings, and end user evaluations have been positive (Rienks and Verbree, 2006). However, the approach has been to browse the meeting content through argumentation diagrams instead of transcripts, and although users experienced that the diagrams made the task easier, they spent more time solving the task.

The purpose of the conversation graphs in this work is to provide users a means to learn the exploitable annotations in the meeting data and to use this knowledge to express meaningful natural language queries to a transcript-based browser. Although users have general knowledge about what argumentation means and what types of argumentative contributions can be made in a meeting (suggestions, agreements, disagreements, etc.), it is not evident how these contributions can be referred to in queries. The conversation graphs are intended to guide users by showing which argumentative categories are available as search criteria in the system.

An important criterion in the design of the graphs is that the visualization of the argumentative annotation has to be intuitive so that users do not need to spend effort on learning the argumentative categories. From this perspective, the graph representation is ideal as it enables us to introduce the intuitive notion of "positive" and "negative" contributions in discussions. Positive contributions are visualized as peaks and negative contributions as valleys along the time axis. Concretely, disagreements are negative arguments and therefore represent low points in the graphs, whereas agreements and decisions are positive and appear at the top. Suggestions are neutral in polarity and are positioned in the middle.

We argue that visualizing argumentative contributions in this positive-negative dimension helps users to distinguish between different argumentative categories easily and to remember them. It enables them to associate the argumentative categories in the positive-negative axis with their prior knowledge about argumentation in real discussions. The cognitive research on memory supports our claim by stating that what we see and remember depends more on what we already know, than on what is actually presented (Cooper, 1998). What is seen and remembered in this case are the argumentative categories "suggestion", "agreement" and "decision" etc. What is already known is that participants of meetings usually discuss by agreeing or opposing each other. What is actually being presented in the graph are lines, dots, colours and labels.

The dots in the graph represent moments where argumentative actions occurred. The unique colours of the dots represent the identification of the speakers. This design works well when there are few participants in the meeting, as it makes it easy to remember which colour belongs to which speaker. In our data there were at most five participants in each meeting. When the number of speakers gets higher, a different design may be required to visualize speakers.

The last piece of information present in the graph is the topical segmentation of the meeting discussion. In our meeting data the segmentation is hierarchical with several levels of sub topics, as can be seen in the example of the topic overview in figure 5.1. We could have included all the sub topics in the conversation graph but would have then faced a problem of readability. The graph would become cluttered with information and it is not evident that the user would be able to use all of it. Therefore in our study, we have chosen to reduce the topical information to only the first-level topics present in the annotation.

We envision that users will use conversation graphs to express natural language queries to the system that combine topical, argumentative and speaker criteria. For example if a user wants to find out what objections (argumentative criterion) the meeting participant Martin (speaker criterion) had about the purpose of the room (topical criterion), the conversation graph shows that Martin disagreed several times during the discussion about that topic. Displaying this information should make it intuitive to search for the relevant meeting episodes by expressing a query in natural language that combines the three criteria, for example ”Show me Martin’s disagreements during the discussion about the purpose of the room”.

The second aspect of how conversation graphs can be useful in meeting data retrieval is that they can help users to browse the results of their search. When a user opens a meeting transcript and browses through the highlighted sections that correspond to their search criteria, they can compare these highlighted sections with the argumentation points in the graph. By referring to the graph the user can extract information about how many sections of the discussion correspond to their search criteria (in our example as many as there are disagreements by Martin in the graph). The user may then derive that some, but not necessarily all, of the search results in the transcript are relevant for answering their original question.

5.3 Research questions and hypotheses

Our first objective in this chapter is to compare topic overviews and conversation graphs in terms of how they enhance natural language search and browsing of annotated meeting discussions. By setting up and executing an appropriate user evaluation, we want to answer four research questions. The first one is:

> **R1.** Are conversation graphs more useful than topic overviews as support for search and browsing?

This question is important because in our design of the system, the use of meeting overviews is optional. Users can always choose to interact with the system directly, without consulting the meeting overview, if they find that the overview does not bring added value to the task. We believe that users can make more use of conversation graphs than standard topic overviews; first, because they visualize multiple annotation layers in the meeting data (layers which are hidden in the user interface), and secondly, because the information is more fine-grained, showing the discussion argument by argument. The hypotheses we have are:

> **H1.** Conversation graphs enhance both querying and browsing. For querying it provides the available search criteria, and for browsing it provides a compact summary of the possible search results.

> **H2.** Topic overviews do not enhance querying. Users are accustomed to content-based search without having any overview of the content. However, topic overviews can enhance browsing in this

domain by providing an order of the topical episodes of meeting discussions.

On the querying level, one of the aims of the meeting overview is to enhance search by helping users to exploit the annotations made on meeting discussions. Our second research question is:

R2. Do meeting overviews influence which search criteria users specify in their queries?

This question is a follow-up of the evaluation in chapter 4, where users had an interactive version of topic overviews in the system. We found that users did not exploit the argumentative annotations as much as we had expected for search of argumentative segments in meeting discussions. We concluded that users did not have sufficient support on how the meetings were annotated to be able to efficiently use them in search. Based on these outcomes, our two hypotheses are:

H1. Topic overviews mainly incent topic and keyword search. Users then read or watch the meeting to decide where argumentative contributions took place.

H2. Conversation graphs incent users to search for speakers and argumentative contributions. The graphs show that instances of these exist in the meeting, and provide a concrete vocabulary for referring to them.

As an additional follow-up of the outcomes of the evaluations in chapter 4, our third research question is:

R3. Do meeting overviews influence how complex queries users formulate?

Here we are particularly interested in whether conversation graphs incent users to express queries linguistically, and to refer to multiple concepts and relations in the domain model. In chapter 4 we found that users who had no meeting overviews to guide their formulation of queries tended to search sequentially with short queries, one search criterion at a time and, that the queries lacked the necessary linguistic structure to disambiguate their domain-specific word meanings. The two hypotheses we have on this question are:

H1. Conversation graphs incent users to query several layers of the meeting data - topical episodes, speakers, and argumentative segments - in a single query. Queries of such complexity are most naturally expressed linguistically, e.g. "Susan's suggestions about the placement of furniture" rather than "Susan, suggestions, furniture".

H2. Topic overviews incent users to query one data layer at a time, starting with the topical one, and to express queries economically, e.g. "colour, sofa" rather than "What colour did they choose for the

> sofa?". This hypothesis is grounded on Grice's maxim of quantity, which states that when humans communicate they make their contributions as informative as required, but not more informative than required (Grice, 1975).

The fourth research question relates to the browsing-aspect of meeting data retrieval.

> **R4.** Do meeting overviews influence how successful users are in interpreting hits in the meeting discussion as answers to questions?

The degree of success in interpreting search results was not addressed directly in chapter 4, only the relevance of search criteria for given search tasks. The interpretation of search results is, however, an important aspect of meeting data retrieval, and one in which meeting overviews may have a potential use. In the design of the Archivus interface, search results can be interpreted by associating the active criteria in the 'Current search criteria' list with the highlighting that these criteria trigger in the meeting books. We do not know to what degree users make these associations while browsing meeting books. Our hypotheses are:

> **H1.** Conversation graphs help users to understand the differences between the types of search criteria that exist in the system (Topic, Speaker, Argumentation) and the precise meaning of their instances in the 'Current search criteria' list. As a consequence, users understand the association between the active criteria and the highlighted sections of the meeting transcripts.
>
> **H2.** Topic overviews incent a keyword-search-thinking. All criteria in the 'Current search criteria' list are interpreted as keywords. The fact that criteria are presented as attribute-value pairs is unintuitive from this perspective. During browsing, the active criteria are expected to appear as explicit words in the transcript, e.g. "suggestion", or "John". When sections of the meeting are highlighted but do not contain the expected keywords, users reject the section as potential answer to their question.

The second objective of this chapter is to evaluate, under strict conditions, the differences between natural language search and menu-based search. In chapter 4, when users had the choice between natural language and tactile interaction, they often chose natural language for selecting criteria from menus. It was not possible to conclude whether natural language querying and menu-selection lead to different search behaviour. In this chapter, we want to make a new evaluation, in which users have access either to natural language or menus. The research question is:

> **R5.** Does the search modality influence what search criteria users specify in their search?

Here we are particularly interested in whether a natural language query interface incents users to specify more search criteria than a menu-based interface, and whether these queries refer to more layers of the meeting annotations than menu-selected criteria. In chapter 4, the results suggested that users who have a tactile interface specify fewer criteria, even to the point where they browse the data directly without any criteria at all. In this chapter we want to validate that result. The hypothesis is:

> **H1.** A natural language query interface incents users to specify more search criteria, and more diverse ones, than a menu-based interface.

If this is the case, then we can provide an argument to the debate about whether or not it is worthwhile to annotate meeting discussions with higher-level annotations. Our argument would be that the degree to which such annotations are exploited in search depends on the search modality, and that the design of the search interface should be taken into account in the design of the meeting database and its data layers.

5.4 Experimental method

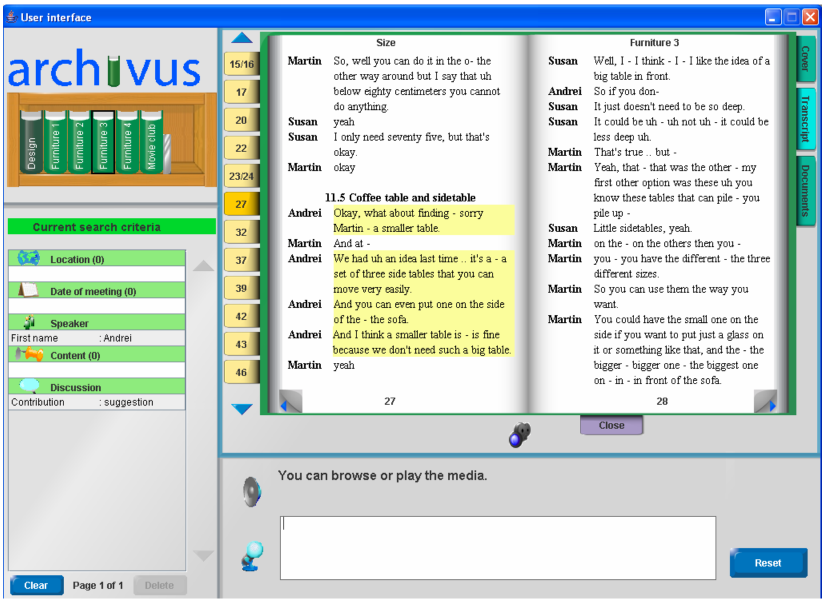
To be able to answer the research questions outlined in 5.3, we designed three evaluation conditions (see section 5.4.1). Two new versions of the Archivus interface were implemented for this purpose (section 5.4.2). The task was defined as a set of questions addressing multiple data-layers in the meeting data (section 5.4.3), and the experimental procedure and documents were updated from the earlier evaluation in chapter 4 (5.4.4). When the experiments were executed, the interactions were again recorded on video and logged by the system. This data was then post-processed to enable us to perform various types of analysis to test our hypotheses (5.4.5).

5.4.1 Evaluation conditions

The evaluation was performed with 30 subjects that were divided into three groups (see table). Two groups had access to a linguistic user interface but received different meeting overviews on sheets of paper. Group 1 received topic overviews and group 2 conversation graphs. There were totally six meetings in the system's database, and six corresponding meeting overviews of each kind. The third group had access to a menu-based version of evaluation conditions and received conversation graphs.

	Linguistic search	**Menu-based search**
Conversation graph	Group 1	Group 3
Topic overview	Group 2	-

Table 5.1: Evaluation conditions



First name : Andrei
Contribution : suggestion

Size

Martin So, well you can do it in the o- the other way around but I say that uh below eighty centimeters you cannot do anything.
Susan yeah
Susan I only need seventy five, but that's okay.
Martin okay

11.5 Coffee table and sidetable

Andrei Okay, what about finding - sorry Martin - a smaller table.
Martin And at -
Andrei We had uh an idea last time .. it's a - a set of three side tables that you can move very easily.
Andrei And you can even put one on the side of the - the sofa.
Andrei And I think a smaller table is - is fine because we don't need such a big table.
Martin yeah

27

Furniture 3

Susan Well, I - I think - I - I like the idea of a big table in front.
Andrei So if you don-
Susan It just doesn't need to be so deep.
Susan It could be uh - uh not uh - it could be less deep uh.
Martin That's true .. but -
Martin Yeah, that - that was the other - my first other option was these uh you know these tables that can pile - you pile up -
Susan Little sidetables, yeah.
Martin on the - on the others then you -
Martin you - you have the different - the three different sizes.
Martin So you can use them the way you want.
Martin You could have the small one on the side if you want to put just a glass on it or something like that, and the - the bigger - bigger one - the biggest one on - in - in front of the sofa.

28

You can browse or play the media.

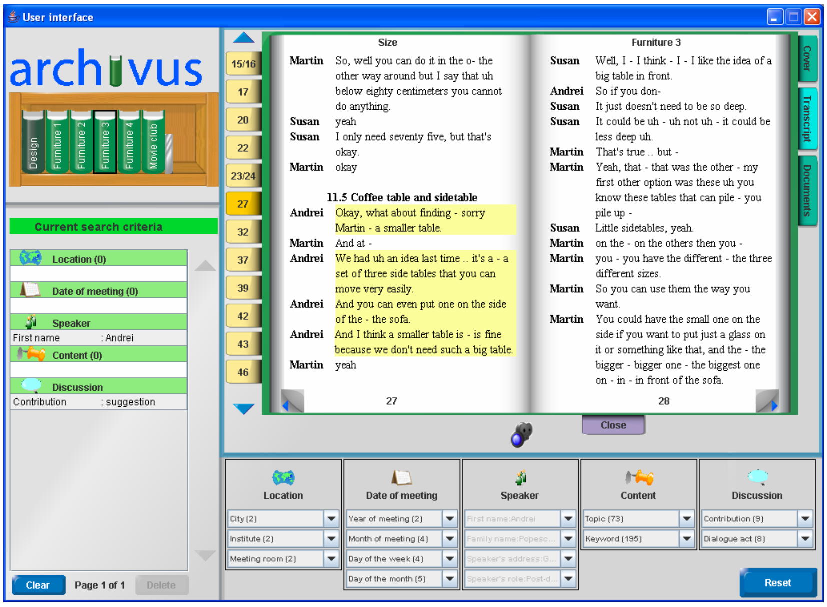
Figure 5.4: Linguistic version of Archivus

Group 1 and group 2 were compared on issues related to meeting overviews (R1-R4). Group 1 and group 3 on the other hand were compared on issues related to search technique (R5). Group 2 was not compared with group 3, as both the user interface and meeting overview differed between these two conditions.

5.4.2 User interfaces

The linguistic version of the Archivus system used in the evaluation is shown in figure 5.4. The main difference between this interface and the language-enabled interface in the previous evaluation is that the search criteria buttons at the bottom of the screen, which allow access to menus, have been removed. In exchange, the Current search criteria list at the left of the screen has been redesigned. Instead of displaying attribute-value pairs in the order that they are specified, this interface divides the list into five sections according to the five types of criteria, and displays the criteria in the appropriate section. For example, when a user makes the request ”Show me Andrei’s suggestions”, the attribute-value pair ”Firstname:Andrei” appears in the Speaker section, and ”Contribution: suggestion” in the Discussion section. The new design is intended to give users a more structured view of the search and, more importantly, give an indication of the scope of natural language queries that the system is able to understand.

The menu-based version of the Archivus system is shown in figure 5.5. Here, the main difference with the earlier version is that the menus are accessible directly, not via search criteria buttons. The new design was motivated by responses in the previous post-evaluation questionnaire. Users thought it was



Size

Martin	So, well you can do it in the o- the other way around but I say that uh below eighty centimeters you cannot do anything.
Susan	yeah
Susan	I only need seventy five, but that's okay.
Martin	okay

11.5 Coffee table and sidetable

Andrei	Okay, what about finding - sorry Martin - a smaller table.
Martin	And at -
Andrei	We had uh an idea last time .. it's a - a set of three side tables that you can move very easily.
Andrei	And you can even put one on the side of the - the sofa.
Andrei	And I think a smaller table is - is fine because we don't need such a big table.
Martin	yeah

27

Furniture 3

Susan	Well, I - I think - I - I like the idea of a big table in front.
Andrei	So if you don-
Susan	It just doesn't need to be so deep.
Susan	It could be uh - uh not uh - it could be less deep uh.
Martin	That's true .. but -
Martin	Yeah, that - that was the other - my first other option was these uh you know these tables that can pile - you pile up -
Susan	Little sidetables, yeah.
Martin	on the - on the others then you -
Martin	you - you have the different - the three different sizes.
Martin	So you can use them the way you want.
Martin	You could have the small one on the side if you want to put just a glass on it or something like that, and the - the bigger - bigger one - the biggest one on - in - in front of the sofa.

28

Figure 5.5: Menu-based version of Archivus

difficult to access criteria hierarchically, e.g. by first choosing the Content button, then either the Topic or Keyword button, and first then the actual menu. With this new design, users do not need to perform as many interaction steps as before to select a criterion. It makes the menu-based and linguistic interfaces more equal in terms of number of interactions with the system to solve a task. Another novelty in the menu-based interface is that the system does not give advice in natural language. In the linguistic version, however, a wizard monitors the system advice, like in the original setup.

5.4.3 Task

The scenario in which users were told to imagine themselves when using the system was the same as in the previous evaluation. They were told to imagine themselves as a new employee who had been asked by the manager to check certain facts about past meetings. The questions in the task were adapted for the current research objectives. From the classification of questions in 4.5.3 only questions that pertained to the topical and argumentative layers of the meeting discussion were chosen. Questions that could be answered without specifying any search criteria, and that only required navigating to the appropriate information in the GUI, were removed. An example is: "*Where was the design meeting held?*". Also questions that addressed referenced documents were removed, e.g. "*How many pictures are there in the Google document?*".

In the current evaluation, the goal was to use questions that addressed the topical, argumentative and speaker identification layers simultaneously, so that it would be possible to study which layers users chose to refer to when specifying search criteria. The questions were formulated in such a way that it was not immediately obvious what the user should say to the system, and which criteria they should specify, e.g. Q5:"*At the end of the meeting, when the participants discussed and gave additional comments about the presentations, they agreed on some furniture pieces. Which ones?*".

The task consisted of 12 questions to be answered during a 30 minute sessions, instead of the previously 40 questions to be answered during two sessions of 20 minutes. We reduced the number of questions because we wanted users to have time to solve all questions. Another reason was that we decided to remove yes-or-no questions and only include short-answer questions. Users were less likely to guess the answers to short-answer questions, but needed more time to find the answer.

The rules for scoring answers to questions were adapted to the new questions. In our set, many questions required browsing to several episodes of the meeting transcript. Therefore we had a five point scale instead of four (see table 5.1).

Score	**Motivation**
1.0	The answer is correct.
0.75	The user found the relevant episode(s) of the meeting discussion, but the answer is incorrect.
0.5	The user found part of the relevant episodes. The answer is incomplete.
0.25	The user specified relevant search criteria but did not find the relevant episode(s) of the discussion. The answer is incorrect.
0	The user did not answer the question.

Table 5.2: Rules for scoring answers to questions in the task

5.4.4 Procedure

The evaluation was again performed as a Wizard of Oz experiment, but only with the linguistic version of the system. The menu-based version was fully automated. The evaluation consisted of three parts:

1. Demographic questionnaire and tutorial (20 min)
2. Evaluation session (30min)
3. Post-evaluation questionnaire (20 min)

There were three versions of the tutorial, one for each condition. The tutorial contained two examples of search scenarios that were aimed to demonstrate the use of the GUI, the different types of search criteria that could be specified, the different ways in which natural language could be used to query the interface (when applicable), and the possible ways in which the meeting overviews could be used as support for search and browsing.

In the post-evaluation questionnaire users were asked to give their opinions about the system in general, the natural language interface in particular, and the meeting overviews that they received on paper.

5.4.5 Data post-processing and analysis

The experiment was recorded on video, and all interactions with the system were automatically logged, to allow for detailed analysis of the recorded data. Three types of post-processing were done on the data.

1. The use of meeting overviews was manually added into the automatically generated logfiles by watching the experiment recordings. When we saw that a user was watching the meeting overview, we added the event to the logfile, placing it before the next interaction with the system.
2. Each interaction with the system was classified as either a search action (the user expressed a query in natural language, or selected a criterion from a menu), or a browsing action (the user opened, read or browsed a meeting transcript, or watched the meeting recording).
3. The search criteria that users specified in their queries were classified as being of type Content (topic or keyword), Speaker, or Argumentation.

The data obtained from the experiment represented approximately 10 hours of experiment recording, 224 solved questions and 6290 interactions with the system.

5.5 Experimental results

5.5.1 Use of meeting overviews

Our first objective of the evaluation was to analyze whether conversation graphs were more useful than topic overviews, and if usefulness was related to search or browsing (**R1**). In both conditions the meeting overviews were given on sheets of paper which means that users could solve the task by exclusively interacting with the system, and not consulting the meeting overview, if they found no use of it.

Results show that users who received conversation graphs (group1) used them more often than users who received topic overviews (group 2)(see figure 5.6). However, users in group 1 consulted their conversation graphs more frequently for some questions in the task than others, whereas users in group 2 consistently chose to solve the task without or with very little help from topic overviews.

To understand better how the meeting overviews were used in the experiment, we compared how often users consulted them before making a natural language query to the system, and how often they looked on them while browsing and reading the meeting transcript. The result we obtained was that in group 1 conversation graphs were used in approximately equal amounts as help for querying and browsing. In group 2 users either did not use the topic overviews at all, or

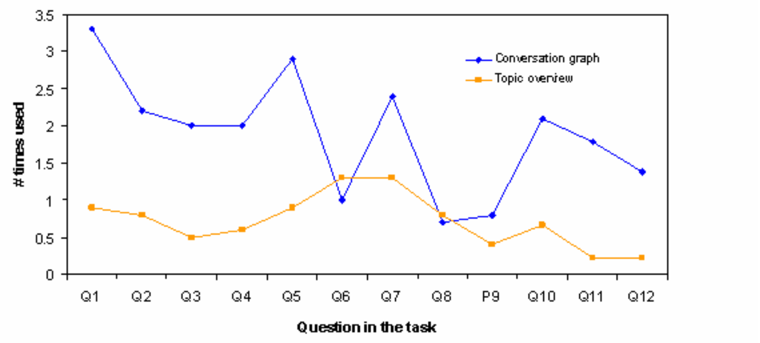


Figure 5.6: Frequency of using meeting overview

they used them only once during a question in the task, and then mainly as an aid for specifying search criteria (see figure 5.7).

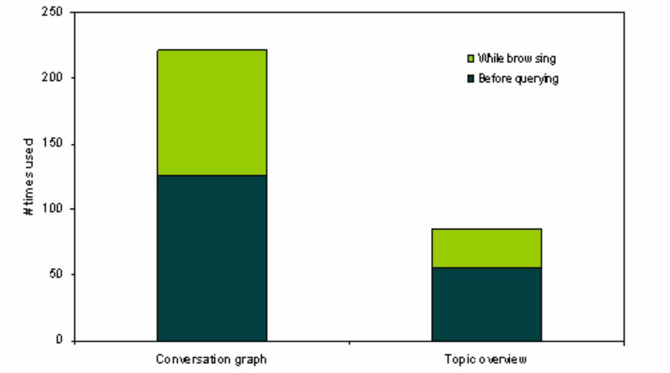


Figure 5.7: Use of meeting overview as support for querying and browsing

In the post-evaluation questionnaire, subjects were asked if and how the meeting overviews were useful to them while solving the task. Group 1 strongly agreed that conversation graphs were useful for multiple purposes, e.g. finding agreement and disagreements, searching for contributions by a specific speaker, and searching for topics. Subjects in group 2 either left the question unanswered or said that the topic overviews were mainly useful for browsing when a topic was discussed more than once during a meeting.

The above results indicate that conversation graphs enhance both querying and browsing of meeting discussions (**H1**). Users get support in specifying search criteria on multiple data-layers of meeting discussions (speakers, argumentation and topics), and the graphs also help when browsing the results of the specified criteria. Our data analysis also confirms that topic overviews enhance neither querying nor browsing (**H2**). This we derive from the fact that users rarely consulted topic overviews during the experiment and gave very few comments about them in the post-evaluation questionnaire.

5.5.2 Exploitation of meeting annotations

In this section we address the question whether meeting overviews influence which search criteria users specify in their queries (**R2**). The hypotheses we want to test are that topic overviews incent users to search only the content-layer of meeting discussions (topics and keywords) (**H1**), and contrary, that conversation graphs incent users to search all three annotation layers (topic, argumentation and speakers) (**H2**).

We counted the number of topic-, speaker-, and argumentation criteria that users specified in the task, and the result was surprising. We could not see any difference at all between the two groups in terms of the ratio. All users specified 56% content criteria, 21% speaker criteria and 23% argumentation criteria. There was a small difference when counting the total number of each criterion, which revealed that users who worked with topic overviews in fact specified more criteria of each type than users who worked with conversation graphs (see figure 5.8).

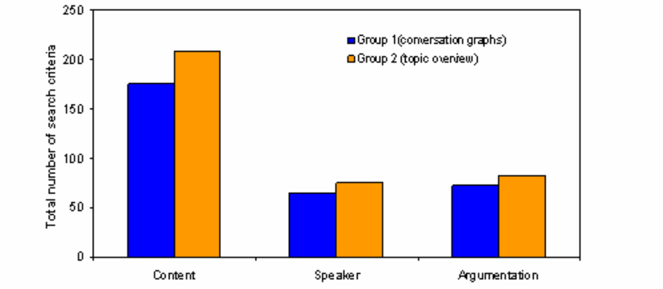


Figure 5.8: Number of content, speaker and argumentation criteria specified in the task (Q1-Q12)

The obtained results speak against our hypothesis that a visual overview of the various annotation layers is needed in order to make users exploit the annotations during search (**H2**). On the other hand, the result does not take into account eventual repetitions of the same query, i.e. that a user specified the same criteria several times while solving a given question. To get a better idea of how frequent repetitions of queries were, we compared the frequency of repetitions in each group. We found that in group 1 (conversation graphs) 8% of queries were repetitions of already specified criteria, whereas in group 2 the number was 20%. When watching the experiment videos we discovered that some users in group 2 would continue to specify search criteria even when they had the relevant piece of transcript highlighted in front of them. Some others started the search by expressing all their search criteria in "one shot" which occasionally lead to an over-constrained situation where no search results were found. In such dead-end situations, some users reacted by repeating the same query over and over instead of clearing the Current search criteria list and starting over. This behaviour suggests that users in group 2 were not as aware of how their queries were interpreted, and what the Current search criteria component of the interface meant. Moreover, in the post-evaluation questionnaire, the opinions about the natural language interaction and selection of search criteria differed between the two groups. Subjects in group 1 reported that when they made a query, the search criteria that appeared in the Current search criteria list reflected precisely what they were searching for. They also found the argumentative criteria very useful as search criteria. Subjects in group 2 on the other hand were more undecided about whether the Current search criteria reflected what they were searching for, and they found the argumentative criteria less useful. Based on these responses, we conclude that users in group 1 specified their search criteria more intentionally, whereas user sin group 2 searched in a more exploratory fashion. Despite the fact that both groups generated the same amount of content-, speaker-, and argumentative criteria, we conclude that conversation graphs influence the process of specifying search criteria: by making explicit how users can exploit the meeting annotations during search and how queries are interpreted by the system.

5.5.3 Complexity of linguistic queries

The third research question addressed in this chapter is whether the form and content of the meeting overview has any influence on query style. In particular, we want to find out if conversation graphs incent users to make more complex linguistic queries that refer to multiple annotation layers on meeting discussions (**R3**).

Here the results were the opposite of what we had expected. Although the differences were not extreme between the groups, we observed that users who worked with conversation graphs tended to express short queries and specify one, or at most two, search criteria at a time (see figure 5.9). 63% of their queries generated one criterion, 25% two criteria, and only 12% three criteria. Also in group 2 the majority of queries generated only one criterion (54%), but users in this group quite frequently also made queries that generated three criteria (26%).

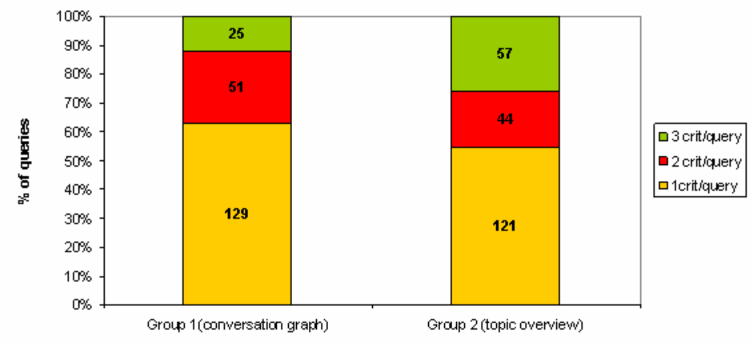


Figure 5.9: Complexity of queries in terms of number of search criteria

In terms of linguistic structure, we did not observe a significant difference in how users expressed queries. In both groups users favoured linguistic querying over keyword-style querying, even when they specified only one search criterion in a query. In group 1 the distribution was somewhat higher with 73% linguistic and 27% keywords, whereas in group 2 the distribution was 66% linguistic and 33% keyword-style. The result is quite different from what we obtained in the previous evaluation with the flexibly multimodal version of Archivus. In the earlier experiment, users had a preference either for linguistic or keyword-style querying, and the number of users in each group was very even. We conclude that the new design of the Archivus interface is more important for triggering linguistic interaction than the form and content of meeting overviews, hence disproving our two hypotheses that conversation graphs incent users to make complex linguistic queries (**H1**) and, that topic overviews incent users to search with keywords (**H2**). What the precise trigger of linguistic querying is needs to be evaluated further, but a possible influencing factor is the new design of the Current search criteria list.

A phenomenon that needs to be mentioned about users' natural language interaction with the system is the "reading questions from card" phenomenon. Some users attempted to solve questions by simply reading the question as it was written in the task. Although there were not so many cases of such queries to the system (8%) the great majority of these queries were made by users in group 2 (topic overviews). Comparing this with answers in the post-evaluation questionnaire, we found that users who had worked with conversation graphs reported more often that the natural language interface allowed them to express themselves in their own terms. These two facts in combination suggest that users who worked with topic overviews were less inclined to express queries in their own terms but instead to follow established patterns, either by copying the formulations of questions in the task or previous examples in the tutorial. Users who worked with conversation graphs on the other hand, seemed more confident about how to express natural language queries, and more inclined to invent their own formulations.

5.5.4 Browsing and interpreting search results

The fourth question in our study is whether conversation graphs are more helpful than topic overviews for browsing and identifying answers to questions in the highlighted sections of the meeting transcripts (**R4**). We have already seen that conversation graphs are used more than topic overviews when users browse meeting data. But to determine how much they actually enhance the browsing, we compare the task scores.

Here we found that there was no strong correlation between the type of meeting overview used in the experiment, and the scores on the task (see figure 5.10). In many questions both groups scored very high. However, group 1 showed a consistently high performance on all the questions, whereas group 2 performed poorly on some questions, for example Q4: " *When Susan made her presentation, Agnes was sceptical about her choice of colours, for example the colour of the sofa. What was her argument against that colour?*". For this question, the difference in mean score was statistically significant with F(1,18)=9.1, Fcrit=4.414, p=0.0073. Users in group 2 simply failed to browse to the relevant episode of the transcript, even when they had the correct search criteria and the relevant meeting episode in their result set.

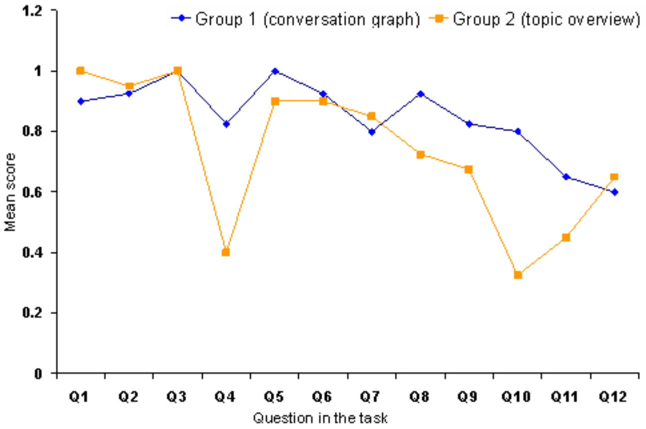


Figure 5.10: Task scores

When comparing the distribution of the five different scores that users could get on a question, we see that group 1 solved more questions with the maximal score (1.0), whereas group 2 more often scored 0.25 or 0 (see figure 5.11). The low scores mean that although users specified relevant search criteria, they did not find the relevant episodes of the meeting discussion. Browsing the meeting data seems to have been generally more difficult for group 2 than for group 1. We attribute the high scores in group 1 to the conversation graphs, which seem to have helped users in the browsing process.

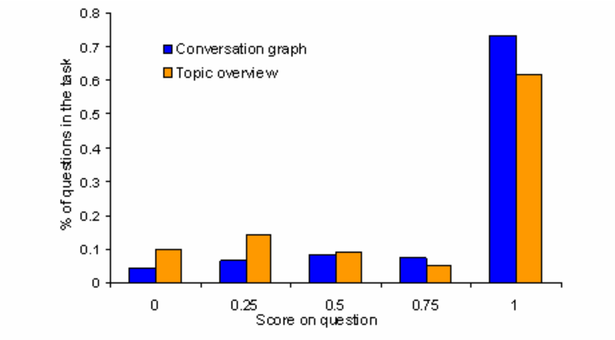


Figure 5.11: Number of questions solved in the task distributed by task score

To explain how conversation graphs enhance browsing, we did observe certain behaviour in group 2 that was less frequent in group 1, namely that users stopped browsing as soon as they had accessed the first highlighted section of the meeting book, and used that first hit as the answer to the question in the task. It is important to mention that even when users have specified all the relevant search criteria, there is never a guarantee that the search results will only contain meeting episodes that are relevant for answering the original question. Meeting discussions are full of contextual information that cannot be captured by any annotation schema, which means that several non-related episodes may be described by exactly the same annotations. For example, if a user searches for episodes where John disagreed about the choice of colour for a sofa, then the results may contain one episode in which John disagreed with the general colour scheme of the room proposed by one meeting participant; and another in which he disagreed with the colour of a given sofa proposed by another meeting participant. Here the user needs to scan the different search results and judge from the context which one is relevant for answering their original question. From our above observations, we conclude that users who work with conversation graphs are more inclined to look into contextual information in the meeting discussion and consequently become very accurate in distinguishing relevant search results from irrelevant ones.

5.5.5 Linguistic and menu-based search

Our final research question concerns the difference between natural language search and menu-based search in terms of how much search criteria users specify and what data-layers they exploit in their search (**R5**). In the flexibly multimodal version of the system in chapter 4, we showed that the natural language interface incented users to specify more search criteria, and more diverse ones, than the menu-based interface. On the other hand, the menu-based interface was cumbersome to use, as the menus were hidden behind search criteria buttons, and had to be selected in a hierarchical procedure. In the current evaluation, the menus are directly accessible as scrollbars, organized by the type of data-layer that the search criteria address.

Our results show that there are very marginal differences between the two groups (see figure 5.12). Users who interacted with the linguistic version of the system specified slightly more content criteria, whereas users in the menu-based version specified more argumentation criteria. The total number of search criteria in the task was almost identical, in average 31-32 search criteria per user.

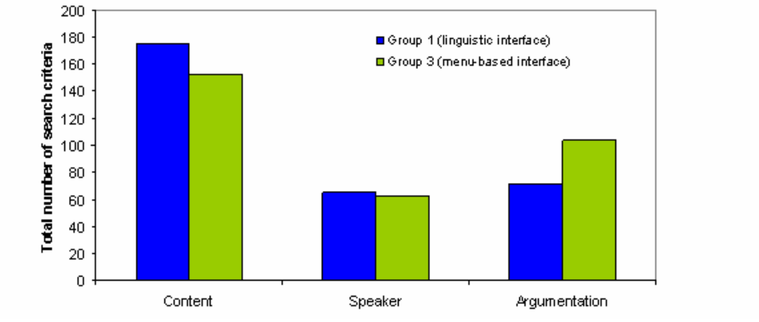


Figure 5.12: Number of content, speaker and argumentation criteria specified in the task (Q1-Q12)

Moreover, when users were asked in the post-evaluation questionnaire how useful they thought that the three types of criteria were, both groups agreed that all three types were useful, and that combining them was useful. These results contradict the previous results with the flexibly multimodal version of the system. There is no longer any difference between the two modalities in terms of search behaviour. Although we may attribute the increased use of search criteria in the menu-based interface to the improved design of the search criteria menus, it is also relevant to note that users in both groups had access to conversation graphs when solving the task. It may be that the availability of conversation graphs in this case contributed to the elevated use of argumentation and speaker criteria in both groups. In any case, our hypothesis that natural language incents users to exploit the meeting annotations more extensively than menu selection is false (**H1**). What we can draw from the present result is that, regardless of interaction modality, it is well worth the effort to annotate meeting discussions with higher-level structures. Users exploit those annotations and find them useful.

5.6 Conclusions of the chapter

The objective of this chapter was to compare two types of meeting overviews to assess their usefulness as support for searching and browsing meeting discussions. Our general conclusion is that conversation graphs are more helpful than topic overviews for finding answers to questions in this domain.

First, we observed that conversation graphs were consulted more often than meeting overviews, and that they were used as support both before expressing a natural language query to the system and as a guide while browsing and reading the meeting transcript. Second, although we did not find proof that conversation graphs incent users to exploit the higher-level meeting annotations more when formulating queries to the system, we did observe that users who worked with conversation graphs specified argumentative and speaker criteria with more intention, and they also found them more useful as search criteria.

In terms of query complexity, we did not find any evidence that conversation graphs incent users to express more elaborative linguistic queries that refer to multiple concepts in the meeting domain. Users generally prefer to search one search criterion at a time. We conclude that sequential search, with progressive refinement of search criteria, is more natural in this domain than trying to minimize the number of searches by specifying all criteria in one query.

Finally, we were able to show that conversation graphs not only enhance querying by giving an explicit scope of what the system can understand, but also enhance browsing, by indicating that a given set of search criteria can generate multiple search results, and that the user needs to look into the context of each result to determine which ones are relevant for answering a given question.

As future work, motivated by responses from users, we would like to integrate conversation graphs and topic overviews as interactive components of the meeting data retrieval system, to be able to compare how they might enhance browsing in that setting. This approach would be an alternative to natural language-based search where users express search criteria. Experimenting with both approaches may give valuable insights about how meeting data is best accessed, searched and browsed in this relatively novel field of information retrieval.

6
Conclusions

The goal of this work was to explore the feasibility and usefulness of implementing a user interface to archived, annotated meetings, which allows users to retrieve answers to complex natural language questions rather than to search for meeting content with plain keywords. Our conclusions relate to three main issues: the argumentative annotation of transcribed meetings, the natural language understanding of questions, and the use of a multimodal graphical user interface to access meeting data. For each issue we summarize our findings and propose directions for future work.

6.1 Meeting annotation

The question whether it is worthwhile to annotate meetings with argumentative categories to make search more efficient is non-trivial. Our initial user requirement analysis showed that such annotations are necessary in order for a system to be able to retrieve answers to typical questions that users have about meetings. On the other hand, when developing and evaluating a prototype of a meeting data retrieval system, we identified two problems. First, argumentative annotation of discussions is difficult, because speakers' contributions in a discussion can be categorized differently depending on the perspective of the annotator. There is no 'neutral' perspective that annotators (or a machine) can take when performing the task. Second, although users are provided with a language-enabled meeting data retrieval system that allows them to ask complex questions about the argumentative aspects of discussions, they do not necessarily take advantage of these features when searching in meetings. On the contrary, users tend to keep to established information retrieval patterns and favour content-based search, or even pure browsing, over any semantic search approaches. However, we have found a means to solve this second problem. When users receive visual overviews of the available meeting annotations - in this case topics, speakers, and argumentative contributions - they change their interaction behaviour. They search in a targeted rather than exploratory manner, they exploit the annotations systematically, and they browse transcripts more efficiently, leading to more successful retrieval of answers to questions. We conclude that argumentative annotations are useful for answering questions on meeting discussions, but only if the annotations are visualized in some explicit manner to users. In this thesis we proposed one such visualization, having the form of colour-coded graphs, but there may be other more appropriate representations. We therefore call for more experimental studies on visualizing data annotations in the context of information search tasks. On one hand, such studies may confirm that visual overviews in general play an important role in making natural language query interfaces usable. On the other hand, studies may reveal that visualizing data annotations can be useful is other more large-scale information search applications, such as the Semantic Web, where web documents are semantically annotated based on domain-ontologies. To date, search engines do apply semantic indexing techniques on documents, but the resulting semantic annotations remain hidden to users.

6.2 Natural language understanding

The issue of natural language understanding (NLU) of questions has a theoretical and a practical side. In theory, i.e. based on example questions elicited from survey studies, we have shown that it is feasible to implement a deep-linguistic natural language understanding engine that interprets complex questions about meeting discussions with relatively few interpretation rules. Our design is based on a clean separation between linguistic and domain-specific processing modules, and therefore represents a flexible environment for extending the interpretation capabilities if needed. For survey-collected questions, the NLU accurately extracts a formal representation of the query based on concepts and relations in the meeting domain model. In practice, however, when users ask questions to a real meeting data retrieval system, the domain-specific interpretation rules are appropriate for only a subset of questions. Users generally do not ask complex questions. Their questions often do not contain sufficient syntactic structure for the NLU engine to make any use of grammatical relations. Even when users are given visual overviews of the meeting, to help them formulate questions about speakers, topics and argumentation, they favour simple questions. We conclude that deep-linguistic NLU technology is not the key requirement for usable natural language interfaces to meeting archives. A much more important factor is the logic of the interaction. Users favour short, sequential questions, because it makes sense to search breadth-first in meeting data and to refine search criteria only if necessary. The risk of asking too specific questions is that the user may end up having no answer. In open-domain question answering the situation is the opposite. Breadth-first search on the internet is not feasible, and there is so much data available that any question is likely to receive some answer. Here the problem is that although deep-linguistic NLU is useful, it is not feasible due to the open domain of questions. General-purpose NLU components simply cannot be designed to perform as accurately as restricted-domain ones. Our work confirms that the real challenge of natural language interfaces is to bridge the gap between the NLU capabilities of the system and users expectations on those capabilities when interacting with the system. The most frequent situation is that users need to adapt to the limitations of the NLU. In our case, we call for studies on how to make users take more advantage of the expressive power of natural language, when the NLU technology is already available.

6.3 Multimodal access to meeting data

The final issue concerns the differences between natural language and menu-based search in meeting data. The reason why we made this comparison is that we believe that a language-enabled GUI is more appropriate than a standard GUI for searching in meeting data. First, language provides an intuitive means for exploiting meeting annotations; second, language can make search more efficient by reducing interaction steps with the GUI; and third, a language interface can enhance the general experience of using a meeting data retrieval system. In our experiments we did find some interesting differences between natural language and menu-based search. For example we found that users tend to specify more search criteria in natural language, and that they are generally more excited about using a speech interface, at least when the speech recognition is almost perfect. However, we also found that the differences subdue when the search task is enhanced with visual meeting overviews. Users then perform equally well, specify equal amount of criteria, and are equally satisfied with the system. The result may seem puzzling at first, but what it really signifies is that the search modality is not the most central issue in the design of meeting data retrieval systems. The way in which the meeting data is presented has a more important impact on how users search the data and how they experience the system as a whole. We conclude that the design of user interfaces for accessing meeting archives should go beyond natural language or menu-based search approaches, and address the use of interactive visual summaries, such as conversation graphs. Experimenting with different combinations of these three approaches may give new insights about how meeting data is best accessed, searched, browsed in this relatively novel field of information retrieval.

Bibliography

Agichtein, E., Burges, C., and Brill, E. (2007). Question answering over implicitly structured web content. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI '07), November 2-5, 2007, Silicon Valley (CA), USA*, pages 18–25, Washington, DC, USA. IEEE Computer Society.

Agrawal, S., Chaudhuri, S., and Das, G. (2002). Dbxplorer: A system for keyword-based search over relational databases. In *Proceedings of the 18th International Conference on Data Engineering (ICDE 2002), February 26 - March 1, 2002, San Jose (CA), USA.*

Ailomaa, M., Melichar, M., Rajman, M., Lisowska, A., and Armstrong, S. (2006). Archivus: a multimodal system for multimedia meeting browsing and retrieval. In *Proceedings of the COLING/ACL on Interactive presentation sessions, July 1721, 2006, Sydney, Australia*, pages 49–52, Morristown, NJ, USA. Association for Computational Linguistics.

Allen, J. (1994). *Natural Language Understanding (2nd Edition).* Addison Wesley.

Allen, J. (1995). *Natural Language Understanding*. Benjamin/Cummings.

Allen, J., Ferguson, G., and Stent, A. (2001a). An architecture for more realistic conversational systems. In *Proceedings of the 6th international conference on Intelligent user interfaces, (IUI '01), January 14 - 17, 2001 Santa Fe (NM), USA*, pages 1–8, New York, NY, USA. ACM.

Allen, J., Byron, D., Dzikovska, M., Ferguson, G., Galescu, L., and Stent, A. (2001b). Toward conversational human-computer interaction. *AI Magazine*, **22**(4), 27–37.

Andr, E., Dybkjaer, L., Minker, W., and Heisterkamp, P., editors (2004). *Affective Dialogue Systems, Tutorial and Research Workshop, ADS 2004, Kloster Irsee, Germany, June 14-16, 2004, Proceedings*, volume 3068 of *Lecture Notes in Computer Science*. Springer.

Andre, E. (2003). Natural language in multimedia/multimodal systems. In R. Mitkov, editor, *Handbook of Handbook of Computational Linguistics*, pages 650–669. Oxford University Press.

Androutsopoulos, L. (1995). Natural language interfaces to databases - an introduction. *Journal of Natural Language Engineering*, **1**, 29–81.

Armstrong, S., Clark, A., Coray, G., Georgescul, M., Pallotta, V., Popescu-Belis, A., Portabella, D., Rajman, M., and Starlander, M. (2003). Natural language queries on natural language data: a database of meeting dialogues. In *Proceedings of the 8th International Conference on Applications of Natural Language to Information Systems (NLDB'2003) June 23-25, 2003, Burg, Germany.*

Aust, H., Oerder, M., Seide, F., and Steinbiss, V. (1995). The philips automatic train timetable information system. *Speech Communication*, **17**(3-4), 249–262.

Austin, J. L. (1962). *How to Do Things With Words*. Oxford University Press.

Baber, C. and Stammers, R. (1989). Is it natural to talk to computers: an experiment using the wizard of oz technique. In E. Megaw, editor, *Contemporary Ergonomics*. Taylor and Francis.

Bachler, M. S., Shum, S. J. B., Roure, D. C. D., Michaelides, D. T., and Page, K. R. (2003). Ontological mediation of meeting structure: Argumentation, annotation, and navigation. In *Proceedings of the 1st International Workshop on Hypermedia and the Semantic Web (HTSW2003), August 26 - 30, 2003, Nottingham, UK*.

Banerjee, S. and Rudnicky, A. I. (2007). Segmenting meetings into agenda items by extracting implicit supervision from human note-taking. In *Proceedings of the 12th international conference on Intelligent user interfaces (IUI '07), January 28-31, 2007, Hawaii, USA*, pages 151–159, New York, NY, USA. ACM.

Banerjee, S., Rose, C., and Rudnicky, A. I. (2005). The necessity of a meeting recording and playback system, and the benefit of topic-level annotations to meeting browsing. In *Proceedings of the International Conference on Human-Computer Interaction (INTERACT 2005) September 12-16, 2005, Rome, Italy*.

Benzmller, C., Fiedler, A., Gabsdil, M., Horacek, H., Kruijff-Korbayov, I., Pinkal, M., Siekmann, J., Tsovaltzi, D., Quoc Vo, B., and Wolska, M. (2003). A wizard of oz experiment for tutorial dialogues in mathematics. In *Proceedings of the 11th International Conference on Artificial Intelligence in Education (AIED 2003), July 20-24, 2003, Sydney, Australia*.

Bernsen, N. and Luz, S. (1999). Smalto: advising interface designers on the use of speech in multimodal systems. In *Proceedings of the IEEE 3rd Workshop on Multimedia Signal Processing, September 13 - 15, 1999, Copenhagen, Denmark*, pages 489–494.

Bernsen, N., Dybkjaer, L., and Dybkjaer, H. (1996). Cooperativity in human-machine and human-human spoken dialogue. *Discourse Processes*, **21(2)**, 213–236.

Bernsen, N., Dybkjaer, L., and Kiilerich, S. (2006). H.c. andersen conversational corpus. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006), May 22-28, Genoa, Italy*.

Bernsen, N. O. (2001). Multimodality in language and speech systems - from theory to design support tool. In B. Granstrm, editor, *Multimodality in Language and Speech Systems.* Dordrecht: Kluwer Academic Publishers.

Bernsen, N. O. and Dybkjaer, L. (1999). A theory of speech in multimodal systems. In *Proceedings of the ISCA Tutorial and Research Workshop on Interactive Dialogue in Multi-Modal Systems, June 22-25, 1999, Kloster Irsee, Germany*, pages 105–108.

Bernsen, N. O. and Dybkjaer, L. (2004). Domain-oriented conversation with h. c. andersen. In *Proceedings of the Workshop on Affective Dialogue Systems (ADS 2004), June 14-16, Kloster Irsee,Germany*, pages 142–153. Springer.

Bhalotia, G., Hulgeri, A., Nakhe, C., Chakrabarti, S., and Sudarshan., S. (2002). Keyword searching and browsing in databases using banks. In *Proceedings of the 18th International Conference on Data Engineering (ICDE 2002), February 26 - March 1, 2002, San Jose (CA), USA.*

Bickmore, T. (2004). Unspoken rules of spoken interaction. *Communications of the ACM*, **47(4)**, 38–44.

Bird, S., Klein, E., Loper, E., and Baldridge, J. (2008). Multidisciplinary instruction with the natural language toolkit. In *Proceedings of the 3rd Workshop on Issues in Teaching Computational Linguistics (TeachCL '08) at the Annual Meeting of the Association for Computational Linguistics (ACL) in Conjunction with the Human Language Technology Conference (ACL 2008 - HLT 2008), June 19-20, 2008, Columbus (OH), USA.*

Blackburn, P. and Bos, J. (2003). Computational semantics. *Theoria*, **18(1)**, 27–45.

Bos, J. (2005). Towards wide-coverage semantic interpretation. In *Proceedings of 6th International Workshop on Computational Semantics (IWCS-6), January 12-14, 2005, Tilburg, The Netherlands*, pages 42–53.

Bouamrane, M. and Luz, S. (2007). Meeting browsing : State-of-the-art review. *Multimedia Systems*, **12**, 439–457.

Boye, J. and Wiren, M. (2008). Robust parsing and spoken negotiative dialogue with databases. *Natural Language Engineering*, **14**(3), 289–312.

Bui, T. H. and Rajman, M. (2004). Rapid Dialogue Prototyping Methodology. Technical report, Swiss Federal Institute of Technology Lausanne (EPFL).

Bui, T. H., Rajman, M., and Melichar, M. (2004). Rapid prototyping methodology. In P. Sojka, I. Kopecek, and K. Pala, editors, *Proceedings of the 7th International Conference on Text, Speech Dialogue (TSD 2004), September 8-11, Brno, Czech Republic*, pages 579–586.

Bunt, H. (1981). Conversational principles in question-answer dialogues. In D. Krallmann and G. Stickel, editors, *Zur Theorie der Frage*, pages 119–141. Narr Verlag.

Bunt, H., Carroll, J., and Satta, G., editors (2004). *New developments in parsing technology*. Number 23 in Text, Speech And Language Technology. Kluwer Academic Publishers, Norwell, MA, USA.

Carroll, J. and Briscoe, T. (2002). High precision extraction of grammatical relations. In *Proceedings of the 19th International Conference on Computational Linguistics, August 24 - Septembre 1, 2002, Taipei, Taiwan*, pages 1–7, Morristown, NJ, USA. Association for Computational Linguistics.

Cenek, P., Melichar, M., and Rajman, M. (2005). A Framework for Rapid Multimodal Application Design. Technical report, Swiss Federal Institute of Technology Lausanne (EPFL).

Chai, J., Lin, J., Zadrozny, W., Ye, Y., Stys-Budzikowska, M., and Horvath, V. (2001). The role of a natural language conversational interface in online sales: A case study. *International Journal of Speech Technology*, **4**, 285–295.

Cheadle, M. and Gamback, B. (2003). Robust semantic analysis for adaptive speech interfaces. *Universal Access in HCI: Inclusive Design in the Information Society*, **4**, 685–689.

Chen, Y., Wang, W., Liu, Z., and Lin, X. (2009). Keyword search on structured and semi-structured data. In *Proceedings of the 35th SIGMOD International Conference on Management of Data (SIGMOD '09), June 29 - July 2, 2009, Providence (RI), USA*, pages 1005–1010, New York, NY, USA. ACM.

Cheng, H., Bratt, H., Mishra, R., Shriberg, E., Upson, S., Chen, J., Weng, F., Peters, S., Cavedon, L., and Niekrasz., J. (2004). A wizard of oz framework for collecting spoken human-computer dialogs. In *Proceedings of the 8th International Conference on Spoken Language Processing (ICSLP '04), October 2004, Jeju Island, Korea*.

Cheon, J. & Grant, M. (2008). A cognitive load approach to metaphorical interface design: Reconsidering theoretical frameworks. In K. McFerrin, editor, *Proceedings of Society for Information Technology and Teacher Education International Conference 2008 (SITE 2008), March 3-7, 2008, Las Vegas (NV), USA*, pages 1054–1059.

Cimiano, P., Haase, P., and Heizmann, J. (2007). Porting natural language interfaces between domains: an experimental user study with the orakel system. In *Proceedings of the 12th International Conference on Intelligent User Interfaces (IUI '07), January 27-31, 2007, Hawaii, USA*, pages 180–189, New York, NY, USA. ACM.

Clark, A. and Popescu-Belis, A. (2004). Multi-level dialogue act tags. In *Proceedings of the 5th SIGDIAL Workshop on Discourse and Dialog (SIGDIAL 04) April 30 - May 1 2004, Cambridge (MA) USA*, pages 163–170.

Cohen, P. R. (1992). The role of natural language in a multimodal interface. In *Proceedings of the 5th Annual ACM Symposium on User interface Software and Technology (UIST '92), November 15-18, 1992, Monterey (CA), USA*, pages 143–149, New York, NY, USA. ACM.

Coletti, P., Cristoforetti, L., Matassoni, M., Omologo, M., Svaizer, P., Geutner, P., and Steffens, F. (2003). A speech driven in-car assistance system. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV 2003), June 9-11, Columbus (OH), USA*, pages 622–626. IEEE.

Comas, P. R. and Turmo, J. (2009). Robust question answering for speech transcripts: Upc experience in qast 2009. In *Working Notes of CLEF 2009*.

Cooper, G. (1998). Research into cognitive load theory and instructional design at unsw. http://education.arts.unsw.edu.au/staff/sweller/clt/index.html retrieved July 2009.

Cox, A. L., Cairns, P. A., Walton, A., and Lee, S. (2008). Tlk or txt? using voice input for sms composition. *Personal Ubiquitous Computing*, **12**(8), 567–588.

Cremers, A. H., Hilhorst, B., and Vermeeren, A. P. (2005). What was discussed by whom, how, when and where? personalized browsing of annotated multimedia meeting recordings. In *Proceedings of the International Conference on Human-Computer Interaction (HCI International 2005), July 22-27, 2005, Las Vegas (NV), USA*, pages 1–10.

Dahlback, N., Jonsson, A., and Ahrenberg, L. (1993). Wizard of oz studies: why and how. In *Proceedings of the 1st International Conference on Intelligent User Interfaces (IUI '93), January 4-7, 1993, Orlando (FL), USA*, pages 193–200, New York, NY, USA. ACM.

Dekleva, S. M. (1994). Is natural language querying practical? *SIGMIS Database*, **25**(2), 24–36.

Delannoy, J. F. (1999). Argumentation mark-up: A proposal. In *Proceedings of the Workshop "Towards Standards and Tools for Discourse Tagging" at the Conference of the Association for Computational Linguistics (ACL'99), June 22, 1999, College Park (MD), USA.*

Dybkjaer, L., Bernsen, N., and Minker, W. (2004). Usability evaluation of multimodal and domain-oriented spoken language dialogue systems. In *Proceedings of the 4th International Conference on Language Resources (LREC 2004), May 26-28, 2004, Lisbon, Portugal.*

Edlund, J., Gustafson, J., Heldner, M., and Hjalmarsson, A. (2008). Towards human-like spoken dialogue systems. *Speech Communication*, **50**(8-9), 630–645.

Erozel, G., Cicekli, N. K., and Cicekli, I. (2008). Natural language querying for video databases. *Information Sciences*, **178**(12), 2534–2552.

Eun, J., Jeong, M., and Lee, G. G. (2005). A multiple classifier-based concept-spotting approach for robust spoken-language understanding. In *Proceedings of the 9th European Conference on Speech, Communication and Technology (Interspeech 2005), September 4-8 2005, Lisbon, Portugal.*

Fellbaum, C., editor (1998). *WordNet: An Electronic Lexical Database (Language, Speech, and Communication).* The MIT Press.

Ferres, D. and Rodrguez, H. (2006). Experiments adapting an open-domain question answering system to the geographical domain using scope-based resources. In *Proceedings of the Workshop on Multilingual Question Answering (MLQA'06) at the 11th Conference of the European Chapter of the Association of Computational Linguistics (EACL 2006), April 3-7, 2006, Trento, Italy.*

Frost, R., A. (2006). Realization of natural language interfaces using lazy functional programming. *ACM Computing Surveys (CSUR)*, **38(4)**, 1–54.

Fujita, K., Nishimoto, K., Sumi, Y., Kunifuji, S., and Mase, K. (1998). Meeting support by visualizing discussion structure and semantics. In *Proceedings of the 2nd International Conference on Knowledge-Based Intelligent Electronic Systems (KES '98), April 21-23, Adelaide, Australia*, volume 1, pages 417–422 vol.1.

Galley, M. and Mckeown, K. (2003). Discourse segmentation of multi-party conversation. In *Proceedings of the 41st Annual Meeting of ACL (ACL 2003), 7-12 July 2003, Sapporo, Japan*, pages 562–569.

Georgescul, M., Clark, A., and Armstrong, S. (2005). Using support vector machines for thematic text segmentation. In *Proceedings of the PASCAL (Pattern Analysis, Statistical Modelling and Computational Learning Network of Excellence) Workshop on Machine Learning, Support Vector Machines, and Large Scale Optimization, March 16- 18, 2005, Thurnau, Germany*.

Glass, J. and Weinstein, E. (2001). Speechbuilder: Facilitating spoken dialogue system development. In *Proceedings of the 7th European Conference on Speech Communication and Technology (EUROSPEECH 2001), September 3-7, Aalborg, Denmark*, pages 1335–1338.

Gordon, T. F. and Karacapilidis, N. I. (1999). The zeno argumentation framework. *Kuenstliche Intelligenz*, **13(3)**, 20–29.

Grasso, M. A., Ebert, D. S., and Finin, T. W. (1998). The integrality of speech in multimodal interfaces. *ACM Transactions on Computer-Human Interaction (TOCHI)*, **5**(4), 303–325.

Grice, H. P. (1975). Logic and conversation. In P. Cole and J. L. Morgan, editors, *Syntax and Semantics volume 3: Speech Acts*, pages 41–58, New York. Academic Press.

Gruenstein, A., Niekrasz, J., and Purver, M. (2005). Meeting structure annotation: Data and tools. In *Proceedings of the 6th SIGdial Workshop on Discourse and Dialogue, September 2-3, 2005, Lisbon, Portugal*, pages 117–127.

Gruenstein, A., McGraw, I., and Badr, I. (2008). The wami toolkit for developing, deploying, and evaluating web-accessible multimodal interfaces. In *Proceedings of the 10th international conference on Multimodal interfaces (IMCI '08), October 20-22, 2008, Chania, Greece*, pages 141–148, New York, NY, USA. ACM.

Hacioglu, K. and Ward, W. (2001). A word graph interface for a flexible concept based speech understanding framework. In *Proceedings of the 7th European Conference on Speech Communication and Technology (EUROSPEECH 2001), September 3-7, Aalborg, Denmark*.

Hallett, C., Scott, D., and Power, R. (2007). Composing questions through conceptual authoring. *Computational Linguistics*, **33**(1), 105–133.

Hasan, B. and Ahmed, M. U. (2007). Effects of interface style on user perceptions and behavioral intention to use computer systems. *Computers in Human Behavior*, **23**(6), 3025–3037.

Henstock, P. V., Lee, Y.-S., Weinstein, C. J., and Pack, D. J. (2001). Toward an improved concept-based information retrieval system. In *The 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '05), August 15-19, 2005, Salvador, Brazil*, pages 384–385.

Hillard, D., Ostendorf, M., and Shriberg, E. (2003). Detection of agreement vs. disagreement in meetings: training with unlabeled data. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (HLT-NAACL '03), May 27 - June 1, 2003, Edmonton, Canada*, pages 34–36, Morristown, NJ, USA. Association for Computational Linguistics.

Hutchins, E. (1995). How a cockpit remembers its speed. *Cognitive Science*, **19**, 265–288.

Ingensand, J. and Golay, F. (2009). Remote-testing techniques for the evaluation of user interaction with web mapping applications. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM GIS 2009), November 4-6, 2009, Seattle (WA) USA.*

Ioannidis, Y. E. and Viglas, S. D. (2006). Conversational querying. *Information Systems*, **31**(1), 33–56.

Jagadish, H. V., Chapman, A., Elkiss, A., Jayapandian, M., Li, Y., Nandi, A., and Yu, C. (2007). Making database systems usable. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data (SIGMOD '07), June 11-14, 2007, Beijing, China*, pages 13–24, New York, NY, USA. ACM.

Janin, A., Ang, J., Bhagat, S., Dhillon, R., Edwards, J., Macias-Guarasa, J., Morgan, N., Peskin, B., Shriberg, E., Stolcke, A., Wooters, C., and Wrede, B. (2004). The icsi meeting project: Resources and research. In *Proceedings of the NIST ICASSP 2004 Meeting Recognition Workshop, May 2004, Montreal.*

Jurafsky, D. and Martin, J. H. (2009). *Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics. 2nd edition.* Prentice-Hall.

Kaiser, E. C., Johnston, M., and Heeman, P. A. (1999). Profer: Predictive, robust finite-state parsing for spoken language. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '99), March 15-19, 1999, Phoenix (AZ), USA*, pages 629–632. IEEE.

Katz, B., Borchardt, G., Felshin, S., and Mora, F. (2007). Harnessing language in mobile environments. In *Proceedings of the 1st IEEE International Conference on Semantic Computing (ICSC 2007), September 17-19, 2007, Irvine (CA), USA.*

Kaufmann, E., Bernstein, A., and Zumstein, R. (2006). Querix: A natural language interface to query ontologies based on clarification dialogs. In *Proceedings of the 5th International Semantic Web Conference (ISWC 2006), November 5 - 9, 2006, Athens (GA), USA*.

Klemmer, S. R., Sinha, A. K., Chen, J., Landay, J. A., Aboobaker, N., and Wang, A. (2000). Suede: A wizard of oz prototyping tool for speech user interfaces. In *Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology (UIST 2000), November 5-8, 2000 San Diego (CA), USA*, pages 1–10.

Kunz, W. and Rittel, H. W. (1970). Issues as elements of information systems. Technical Report 013, Universitaet Stuttgart, Institut fuer Grundlagen der Planung.

Lalanne, D., Ingold, R., vonRotz, D., Behera, A., Mekhaldi, D., and Popescu-Belis, A. (2004). Using static documents as structured and thematic interfaces to multimedia meeting archives. In H. Bourlard and S. Bengio, editors, *Multimodal Interaction and Related Machine Learning Algorithms*, pages 87–100. Springer.

Larsson, S. (2002). Issues under negotiation. In *Proceedings of the 3rd SIGdial Workshop on Discourse and Dialogue, July 11-12, 2002, Philadelphia (PA), USA*, pages 103–112, Morristown, NJ, USA. Association for Computational Linguistics.

Lee, M. and Billinghurst, M. (2008). A wizard of oz study for an ar multimodal interface. In *Proceedings of the 10th international conference on Multimodal interfaces (IMCI '08), October 20-22, 2008, Chania, Greece*, pages 249–256, New York, NY, USA. ACM.

Leidner, J. L. (2005). A wireless natural language search engine. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '05), August 15-19, 2005, Salvador, Brazil*, pages 677–677, New York, NY, USA. ACM.

Linckels, S., Repp, S., Karam, N., and Meinel, C. (2007). The virtual tele-task professor: semantic search in recorded lectures. In *Proceedings of the 38th SIGCSE technical symposium on Computer science education (SIGCSE '07), March 7-10, 2007, Covington, Kentucky, USA*, pages 50–54, New York, NY, USA. ACM.

Lisowska, A. (2003). Multimodal interface design for the multimodal meeting domain: Preliminary indications from a query analysis study. project report im2.mdm-11. Technical report, University of Geneva, Geneva, Switzerland.

Lisowska, A. (2007). *Multimodal Interface Design for Multimodal Meeting Content Retrieval*. Ph.D. thesis, Multilingual Information Processing Department, School of Translation and Interpretation, University of Geneva, Geneva, Switzerland.

Lisowska, A., Popescu-Belis, A., and Armstrong, S. (2004). User query analysis for the specification and evaluation of a dialogue processing and retrieval system. In *Proceedings of the 4th International Conference on Language Resources (LREC 2004), May 26-28, 2004, Lisbon, Portugal*, page 993996.

Lisowska, A., Armstrong, S., Betrancourt, M., and Rajman, M. (2007). Minimizing modality bias when exploring input preferences for multimodal systems in new domains: the archivus case study. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '07), 28 April - 3 May, 2007, San Jose (CA), USA*, pages 1805–1810, New York, NY, USA. ACM.

Lisowska, A., Melichar, M., Ailomaa, M., and Rajman, M. (2009). Designing and running large-scale woz experiments for multimodal systems: Practical advice, pitfalls and happy endings. *Human-Computer Interaction: A Journal of Theoretical, Empirical, and Methodological Issues of User Science and of System Design (submitted).*

Lopez, V., Pasin, M., and Motta, E. (2005). Aqualog: An ontology-portable question answering system for the semantic web. In *Proceedings on the 2nd European Semantic Web Conference (ESWC 2005), May 29 - June 1, 2005, Heraklion, Greece*.

McTear, M. F. (2002). Spoken dialogue technology: enabling the conversational user interface. *ACM Computing Surveys (CSUR)*, **34**(1), 90–169.

Melichar, M. (2008). *Design of multimodal dialogue-based systems*. Ph.D. thesis, Swiss Federal Institute of Technology, Lausanne (EPFL), Lausanne.

Melichar, M., Cenek, P., Ailomaa, M., Lisowska, A., and Rajman, M. (2006). From Vocal to Multimodal Dialogue Management. In *Proceedings of the 8th International Conference on Multimodal Interfaces (ICMI06), November 2-4, 2006, Banff, Canada*.

Michaelides, D., Buckingham Shum, S., Juby, B., Mancini, C., Slack, R., Bachler, M., Procter, R., Daw, M., Rowley, A., Chown, T., De Roure, D., and Hewitt, T. (2006). Memetic: Semantic meeting memory. In *Proceedings of the 15th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, (WETICE 2006), June 26-28, 2006, Manchester, UK*, pages 382–387.

Milward, D. (2000). Distributing representation for robust interpretation of dialogue utterances. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics (ACL '00), October 3-6, 2000, Hong Kong, China*, pages 133–141, Morristown, NJ, USA. Association for Computational Linguistics.

Moeller, S., Krebber, J., Raake, A., Smeele, P., Rajman, M., Melichar, M., Pallotta, V., Tsakou, G., Kladis, B., Vovos, A., Hoonhout, J., Schuchardt, D., Fakotakis, N., Ganchev, T., and Potamitis, I. (2004). INSPIRE: Evaluation of a Smart-Home System for Infotainment Management and Device Control. In *Proceedings of the 4th International Conference on Language Resources (LREC 2004), May 26-28, 2004, Lisbon, Portugal*, pages 1603–1606.

Molla, D. and Vicedo, J. L. (2007). Question answering in restricted domains: An overview. *Computational Linguistics*, **33**(1), 41–61.

Nakao, Y., Rayner, M., Chatzichrisafis, N., Isahara, H., Bouillon, P., Hockey, B. A., and Kanzaki, K. (2006). Recent usability improvements in medslt: Back-translation & help system. In *Proceedings of the Natural Language Processing Conference (NLP2006), March 14-16, 2006, Tokyo, Japan.*

Neto, A. T., Bittar, T. J., Fortes, R. P. M., and Felizardo, K. (2009). Developing and evaluating web multimodal interfaces - a case study with usability principles. In *Proceedings of the 2009 ACM Symposium on Applied Computing (SAC '09), March 9-12, 2009, Honolulu, Hawaii, USA*, pages 116–120, New York, NY, USA. ACM.

Nielsen, J. (1995). Usability inspection methods. In *Proceedings of the Conference companion on Human factors in computing systems (CHI '95), May 7 - 11, 1995, Denver (CO), USA*, pages 377–378, New York, NY, USA. ACM.

Nijholt, A., Akker, d., and Heylen, D. (2006). Meetings and meeting modeling in smart environments. *Artificial Intelligence and Society*, **20**(2), 202–220.

Niu, Y. and Hirst, G. (2004). Analysis of semantic classes in medical text for question answering. In *Proceedings of the Workshop on Question Answering in Restricted Domains at the 42nd Annual Meeting of the Association for Computational Linguistics (ACL 2004), July 21-26, 2004, Barcelona, Spain*, pages 54–6.

Norgaard, M. and Hornbaek, K. (2006). What do usability evaluators do in practice?: an explorative study of think-aloud testing. In *Proceedings of the 6th conference on Designing Interactive systems (DIS '06), June 26-28 2006, University Park (PA), USA*, pages 209–218, New York, NY, USA. ACM.

Norman, D. A. (1991). Cognitive artifacts. In J. M. Carroll, editor, *Designing interaction: psychology at the human-computer interface*, pages 17–38, New York, NY, USA. Cambridge University Press.

Olsen, D. R. (1999). Interacting in chaos. *Interactions*, **6**(5), 42–54.

Oviatt, S. (2003). Multimodal interfaces. In J. Jacko and A. Sears, editors, *The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications*, pages 286–304. L. Erlbaum Associates Inc.

Oviatt, S., Darves, C., and Coulston, R. (2004). Toward adaptive conversational interfaces: Modeling speech convergence with animated personas. *ACM Transactions on Computer-Human Interaction (TOCHI)*, **11**(3), 300–328.

Oviatt, S. L. (1999). Ten myths of multimodal interaction. *Communications of the ACM*, **42(11)**, 74–81.

Pallotta, V. and Ghorbel, H. (2003). Argumentative segmentation and annotation guide-lines. Technical report, Swiss Federal Institute of Technology Lausanne (EPFL).

Pallotta, V., Ghorbel, H., Ruch, P., and Coray, G. (2004). An argumentative annotation schema for meeting discussions. In *Proceedings of the 4th International Conference on Language Resources (LREC 2004), May 26-28, 2004, Lisbon, Portugal*, pages 1003–1006.

Pallotta, V., Seretan, V., and Ailomaa, M. (2007). User requirement analysis for meeting information retrieval based on query elicitation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL 2007), June 25-27, Prague, Czech Republic*, pages 1008–1015. Association for Computational Linguistics.

Pearson, J., Hu, J., Branigan, H. P., Pickering, M. J., and Nass, C. I. (2006). Adaptive language behavior in hci: how expectations and beliefs about a system affect users' word choice. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '06), April 24-27 2006, Montral (QC), Canada*, pages 1177–1180, New York, NY, USA. ACM.

Popescu, A.-M., Etzioni, O., and Kautz, H. (2003). Towards a theory of natural language interfaces to databases. In *Proceedings of the 8th international conference on Intelligent user interfaces (IUI '03), January 12-15, 2003, Miami (FL), USA*, pages 327–327, New York, NY, USA. ACM.

Popescu-Belis, A. and Gorgescul, M. (2006). Tqb: Accessing multimedia data using a transcript-based query and browsing interface. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006), May 22-28, Genoa, Italy*, pages 1560–1565.

Popescu-Belis, A., Clark, A., Georgescul, M., Zufferey, S., and Lalanne, D. (2004). Shallow dialogue processing using machine learning algorithms (or not). In B. S. and B. H., editors, *Proceedings of the Workshop on Machine Learning for Multimodal Interaction (MLMI 2004)*, pages 277–290. Springer-Verlag, Berlin.

Prager, J. M. (2006). Open-domain question-answering. *Foundations and Trends in Information Retrieval*, **1**(2), 91–231.

Qvarfordt, P., Jonsson, A., and Dahlback, N. (2003). The role of spoken feedback in experiencing multimodal interfaces as human-like. In *Proceedings of the 5th international conference on Multimodal interfaces (ICMI '03), November 5-7, 2003, Vancouver (BC), Canada.*, pages 250–257, New York, NY, USA. ACM.

Rajman, M., Ailomaa, M., Lisowska, A., Melichar, M., and Armstrong, S. (2006). Extending the Wizard of Oz Methodology for Language-enabled Multimodal Systems. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006), May 22-28, Genoa, Italy*.

Ramachandran, V. A. and Krishnamurthi, I. (2009). Nlion: Natural language interface for querying ontologies. In *Proceedings of the 2nd Bangalore Annual Compute Conference (COMPUTE '09), January 9-10, 2009, Bangalore, India*, pages 1–4, New York, NY, USA. ACM.

Rauschert, I., Agrawal, P., Sharma, R., Fuhrmann, S., Brewer, I., and MacEachren, A. (2002). Designing a human-centered, multimodal gis interface to support emergency management. In *Proceedings of the 10th ACM International Symposium on Advances in Geographic Information Systems (ACM-GIS 2002), November 8-9, 2002, McLean (VA), USA*, pages 119–124, New York, NY, USA. ACM.

Rayner, M., Bouillon, P., Hockey, B. A., Chatzichrisafis, N., and Starlander, M. (2004). Comparing rule-based and statistical approaches to speech understanding in a limited domain speech translation system. In *Proceedings of the 10th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-2004), October 4-6, 2004, Baltimore (MD), USA*.

Rayner, M., Bouillon, P., Flores, G., Ehsani, F., Starlander, M., Hockey, B. A., Brotanek, J., and Biewald, L. (2008). A small-vocabulary shared task for medical speech translation. In *Proceedings of Coling 2008 Workshop on Speech Processing for Safety Critical Translation and Pervasive Applications, August 23, 2008, Manchester, UK*.

Rienks, R. and Verbree, D. (2006). About the usefulness and learnability of argument-diagrams from real discussions. In *Proceedings of the 3rd International Machine Learning for Multimodal Interaction Workshop (MLMI 2006), May 1-4, 2006, Bethesda (MD), USA*.

Rienks, R., Heylen, D., and van der Weijden, E. (2005). Argument diagramming of meeting conversations. In A. Vinciarelli and J.-M. Odobez, editors, *Proceedings of the Multimodal Multiparty Meeting Processing Workshop at the 7th International Conference on Multimodal Interfaces (ICMI 2005) October 3-7, 2005, Trento, Italy*, pages 85–92. Imported from HMI.

Rosenfeld, R., Olsen, D., and Rudnicky, A. (2001). Universal speech interfaces. *Interactions*, **8**(6), 34–44.

Sagae, K., Lavie, A., and MacWhinney, B. (2003). Combining rule-based and data-driven techniques for grammatical relation extraction in spoken langugage. In *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT-2003), April 23-25, Nancy, France*, pages 153–162.

Salber, D. and Coutaz, J. (1993). Applying the wizard of oz technique to the study of multimodal systems. In *Proceedings of the East-West International Conference on Human-Computer Interaction (EWHCI '93), August 3-7, 1993, Moscow, Russia*, pages 219–230, London, UK. Springer-Verlag.

Searle, J. (1969). *Speech Acts*. Cambridge University Press.

Sears, A. and Jacko, J. A., editors (2007). *The human-computer interaction handbook.* L. Erlbaum Associates.

Seneff, S. (1992). Tina: a natural language system for spoken language applications. *Computational Linguistics*, **18**(1), 61–86.

Shneiderman, B. (1980). *Software psychology: Human factors in computer and information systems (Winthrop computer systems series)*. Winthrop Publishers.

Shneiderman, B. (2000). The limits of speech recognition. *Communications of the ACM*, **43**(9), 63–65.

Shneiderman, B. and Plaisant, C. (2009). *Designing the User Interface: Strategies for Effective Human-Computer Interaction (4th Edition)*. Pearson Addison Wesley.

Shriberg, E., Dhillon, R., Bhagat, S., Ang, J., and Carvey, H. (2004). The icsi meeting recorder dialog act (mrda) corpus. In *Proceedings of HLT-NAACL SIGDIAL Workshop, April-May, 2004, Boston (MA), USA*.

Sidner, C. (1997). Creating interfaces founded on principles of discourse communication and collaboration. In N. R. Council, editor, *More than screen deep: toward every-citizen interfaces to the nations information infrastructure*, pages 515–321. National Academy press: Washington, D.C.

Skantze, G. (2005). Galatea: A discourse modeller supporting concept-level error handling in spoken dialogue systems. In *Proceedings of the 6th SIGdial Workshop on Discourse and Dialogue (SIGDIAL 2005) September 2-3, 2005, Lisbon, Portugal*, pages 178–189.

Sporka, A. J. and Slavik, P. (2008). Vocal control of a radio-controlled car. *ACM SIGACCESS Accessibility and Computing*, **91**(91), 3–8.

Stanciulescu, A., Limbourg, Q., Vanderdonckt, J., Michotte, B., and Montero, F. (2005). A transformational approach for multimodal web user interfaces based on usixml. In *Proceedings of the 7th International Conference on Multimodal Interfaces (ICMI '05), October 3-7, 2005, Trento, Italy*, pages 259–266, New York, NY, USA. ACM.

Stolcke, A., Ries, K., Coccaro, N., Shriberg, E., Bates, R., Jurafsky, D., Taylor, P., Martin, R., Van, C., and Meteer, E.-D. M. (2000). Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, **26**, 339–373.

Stone, M. (2004). Intention, interpretation and the computational structure of language. *Cognitive Science*, **5**, 781–809.

Sturm, J., Bakx, I., Cranen, B., Terken, J., and Wang, F. (2002). The effect of prolonged use on multimodal interaction,. In *Proceedings of the ISCA Tutorial and Research Workshop on Interactive Dialogue in Multi-Modal Systems, June 22-25, 1999, Kloster Irsee, Germany*.

Tartir, S., McKnight, B., and Arpinar, I. B. (2009). Semanticqa: web-based ontology-driven question answering. In *Proceedings of the 2009 ACM symposium on Applied Computing (SAC '09), March 9-12, 2009, Honolulu, Hawaii, USA*, pages 1275–1276, New York, NY, USA. ACM.

Thomson, D. and Wisowaty, J. (1999). User confusion in natural language services. In *Proceedings of the ISCA Tutorial and Research Workshop on Interactive Dialogue in Multi-Modal Systems, June 22-25, 1999, Kloster Irsee, Germany*, pages 189–196.

Tomko, S., Harris, T. K., Toth, A., Sanders, J., Rudnicky, A., and Rosenfeld, R. (2005). Towards efficient human machine speech communication: The speech graffiti project. *ACM Transactions on Speech and Language Processing (TSLP)*, **2**(1), 2.

Traum, D. (2003). Semantics and pragmatics of questions and answers for dialogue agents. In H. Bunt, editor, *Proceedings of the 5th International Workshop on Computational Semantics (IWCS-5), January 15-17, 2003, Tilburg, The Netherlands*, pages 380–394.

Traum, D., Swartout, W., Gratch, J., Marsela, S., Kenney, P., Hovy, E., Narayanan, S., Fast, E., Martinovski, B., Baghat, R., Robinson, S., Marshall, A., Wang, D., Gandhe, S., and Leuski, A. (2005). Virtual humans for non-team interaction training. In *Proceedings of the 6th SIGdial Workshop on Discourse and Dialogue (SIGDIAL 2005) September 2-3, 2005, Lisbon, Portugal*.

Turmo, J., Comas, P. R., Rosset, S., Galibert, O., Moreau, N., Mostefa, D., Rosso, P., and Buscaldi, D. (2009). Overview of qast 2009. In *Working Notes of CLEF 2009*.

Verbree, A. (2006). *On the structuring of discussion transcripts based on utterances automatically classified*. Master's thesis, University of Twente, The Netherlands.

Vinciarelli, A. (2005). Noisy text categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **27**(12), 1882–1895.

Voorhees, E. M. (2001). The trec question answering track. *Natural Language Engineering*, **7**(4), 361–378.

Voorhees, E. M. and Buckland, L. P., editors (2007). *NIST Special Publication 500-274: Proceedings of the Sixteenth Text Retrieval Conference (TREC 2007), November 5-9, 2007, Gaithersburg MD, USA*.

Walker, M. and Whittaker, S. (1989). When natural language is better than menus: A field study . Technical Report HPL-BRC-TR-89-020, HP Laboratories, Bristol, England.

Walker, M. A., Fromer, J., Di Fabbrizio, G., Mestel, C., and Hindle, D. (1998). What can i say?: evaluating a spoken language interface to email. In *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '98) April 18-23, Los Angeles (CA), USA*, pages 582–589, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.

Wang, Y.-Y. and Acero, A. (2005). Sgstudio: Rapid semantic grammar development for spoken language understanding. In *Proceedings of the 9th European Conference on Speech Communication and technology (Interspeech2005), September 4-8, 2005, Lisbon, Portugal*.

Ward, W. (1989). Understanding spontaneous speech. In *Proceedings of the Workshop on Speech and Natural Language (HLT '89), October 15-18, 1989, Cape Cod (MA), USA*, pages 137–141, Morristown, NJ, USA. Association for Computational Linguistics.

Watson, R., Carroll, J., and Briscoe, T. (2005). Efficient extraction of grammatical relations. In *Proceedings of the 9th International Workshop on Parsing Technologies, October 9-10, 2005, Vancouver (BC), Canada*.

Wellner, P., Flynn, M., and Guillemot, M. (2004). Browsing recordings of multi-party interaction in ambient intelligent environments. In *Proceedings of the Workshop on "Lost in Ambient Intelligence" at the Conference on Human Factors in Computing Systems (CHI 2004), April 24-29, 2004, Vienna, Austria*.

Wellner, P., Flynn, M., Tucker, S., and Whittaker., S. (2005). A meeting browser evaluation test. In *Proceedings of International Conference on Human Factors in Computing Science (CHI 2005), April 2-7, 2005, Portand (OR) USA*.

Wiren, M., Eklund, R., Engberg, F., and Westermark, J. (2007). Experiences of an in-service wizard-of-oz data collection for the deployment of a call-routing application. In *Proceedings of the Workshop on Bridging the Gap at the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT '07), April 22-27, 2007, Rochester (NY), USA*, pages 56–63, Morristown, NJ, USA. Association for Computational Linguistics.

Woods, W., Kaplan, R., and Webber., B. (1972). The lunar sciences natural language information system: Final report. Technical Report BBN Report 2378, Bolt Beranek and Newman Inc., Cambridge (MA), USA.

Zue, V. and Glass, J. (2000). Conversational interfaces: advances and challenges. *Proceedings of the IEEE*, **88**(8), 1166–1180.

Zue, V., Seneff, S., Glass, J., Polifroni, J., Pao, C., Hazen, T. J., and Hetherington, L. (2000). Jupiter: A telephone-based conversational interface for weather information. *IEEE Transactions on Speech and Audio Processing*, **8**, 85–96.

Curriculum Vitae

Marita Ailomaa

Personal details

Born:	16.06.1980
Nationality:	Swedish
Marital status:	Married, one child, born 6.9.2008

Education

2004-2009	PhD in Computer Science Ecole Polytechnique Fédérale de Lausanne
2000-2004	Master of Arts in Computational Linguistics University of Göteborg, Sweden
1996-1999	High school Graduation, the Science Program Mölnlycke, Sweden

Technical skills

Programming languages: Java, Prolog, PHP, Javascript, C, Perl, Shell
Multimedia: XML, SVG, XSLT
Databases: PostgreSQL, mySQL, object-oriented databases
Operating systems: Windows, Mac, Unix
Office software: MSOffice, OpenOffice, Latex
Webmaster: http://globalcomputing.epfl.ch/

Technical experience

Building language technology and resources: Semantic lexica, syntactic grammars, word sense disambiguation, natural language understanding, natural language interfaces to databases, dialogue systems

Evaluation of system prototypes: User requirement analysis, speech interface evaluation, graphical interface evaluation, advanced simulation of speech recognition software, evaluation of information search applications, evaluation of multimedia applications, information visualization; evaluation of usability, user performance and user satisfaction

Language skills

Swedish, Finnish	Native
English, German, Swiss-German, French	Fluent
Italian	Basic

Teaching skills

2006-2009	Teaching assistant in Computational Linguistics, a master-level course in computer and communication sciences at the EPFL
2007	Attended the course Tutoring skills offered to PhD students with teaching duties at the EPFL
2000	Private support teacher in Maths and German to 13-year old student with dyslexia

Team work experience

In the context of my PhD I have worked in a team of three researchers to define research goals, specify, set up and execute user evaluations of system prototypes, write reports on the results and present them at conferences and venues.

Publications

Conference papers

Ailomaa, M., and Rajman, M. (2009) Enhancing natural language search in meeting data with visual meeting overviews. Proceedings of the 10th Annual Conference of the NZ ACM Special Interest Group on Human-Computer Interaction (CHINZ 2009), Auckland, New Zealand, 6-7 July.

Pallotta, V., Seretan, V., and Ailomaa, M. (2007) User Requirement Analysis for Meeting Information Retrieval Based on Query Elicitation. Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL 2007), Prague, Czech Republic, 25-27 June, pp. 1008-1015.

Ailomaa, M., Lisowska, A., Melichar, M., Armstrong, S., and Rajman, M. (2006) Archivus: A multimodal system for multimedia meeting retrieval and browsing. Proceedings of the COLING/ACL Interactive Presentations Sessions, Sydney, Australia, 17-21 July, pp. 49-52.

Melichar, M., Cenek, P., Ailomaa, M., and Lisowska, A. (2006) From Vocal to Multimodal Dialogue Management. Eighth International Conference on Multimodal Interfaces (ICMI'06), Banff, Canada, 2-4 November, pp. 59-67

Rajman, M., Ailomaa, M., Lisowska, A., Melichar, M., and Armstrong, S. (2006) Extending the Wizard of Oz methodology for Language-Enabled Multimodal Systems. Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006), Genoa, Italy, 22-28 May, pp. 2539-2543.

Ailomaa, M., Kadlec, V., Rajman, M., and Chappelier, J-C. (2005) Robust Stochastic Parsing: Comparing Two Approaches for Processing Extra-Grammatical Sentences. Proceedings of the 15th NODALIDA conference, Joensuu, Finland, 20-21 May, pp. 1-7.

Ailomaa, M. Kadlec, V., Rajman, M., and Chappelier, J-C. (2005) Efficient Processing of Extra-Grammatical Sentences: Comparing and Combining Two Approaches to Robust Stochastic Parsing. Proceedings of the International Symposium on Applied Stochastic Models and Data Analysis (ASMDA 2005), pp. 81-88.

Kadlec, V. and Ailomaa, M. and Chappelier, J.-C. and Rajman, M.(2005) Robust Stochastic Parsing using Optimal Maximum Coverage. Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2005), pp. 258-263.

Papers and abstracts for poster presentations and demos

Ailomaa, M., and Lisowska, A. (2007) Archivus: A user performance analysis with speech, keyboard and mouse as interaction modalities. Poster at the 4th Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms (MLMI 2007), Brno, Czech republic, 28-30 June.

Cenek, P., Melichar, M., Lisowska, A., and Ailomaa, M. (2006) Archivus: A multimodal system for multimedia meeting browsing and retrieval. Demonstration at Text, Speech and Dialogue conference (TSD'06), Brno, Czech Republic, 11-15 September.

Ghorbel, H., Ailomaa, M., and Rajman, M. (2005) Answering natural language queries on spoken dialogs in meeting discussions. Poster presented at the 2nd Joint Conference on Multimodal Interaction and Related Machine Learning Algorithms, Edinbourgh, UK, 11-13 July.

Technical reports

Pallotta, V., Seretan, V., and Ailomaa, M. (2006) Query types in the meeting domain: assessing the role of argumentative structure in answering questions on meeting discussion records. Technical report No. LIA-REPORT-2009-001, Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland.

Ailomaa, M., and Rajman, M. (2005) Natural Language Techniques for Model-Driven Semantic Constraint Extraction. Technical report No. LIA-REPORT-2007-001, Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland.

Ailomaa, M. (2004) Two Approaches to Robust Stochastic Parsing. Master's Thesis. Technical Report No. 200497. Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland.